J Intell Manuf
DOI 10.1007/s10845-017-1350-2

Review of job shop scheduling research and its new perspectives under Industry 4.0

Jian Zhang[1] · Guofu Ding[1] · Yisheng Zou[1] · Shengfeng Qin[2] · Jianlin Fu[1]

Received: 13 October 2016 / Accepted: 30 July 2017

Abstract Traditional job shop scheduling is concentrated on centralized scheduling or semi-distributed scheduling. Under the Industry 4.0, the scheduling should deal with a smart and distributed manufacturing system supported by novel and emerging manufacturing technologies such as mass customization, Cyber-Physics Systems, Digital Twin, and SMAC (Social, Mobile, Analytics, Cloud). The scheduling research needs to shift its focus to smart distributed scheduling modeling and optimization. In order to transferring traditional scheduling into smart distributed scheduling (SDS), we aim to answer two questions: (1) what traditional scheduling methods and techniques can be combined and reused in SDS and (2) what are new methods and techniques required for SDS. In this paper, we first review existing researches from over 120 papers and answer the first question and then we explore a future research direction in SDS and discuss the new techniques for developing future new JSP scheduling models and constructing a framework on solving the JSP problem under Industry 4.0.

✉ Yisheng Zou
zysapple@sina.com

Jian Zhang
jerrysmail@263.net

Guofu Ding
dingguofu@163.com

Shengfeng Qin
sheng-feng.qin@northumbria.ac.ubk

Jianlin Fu
jlfuhappy@home.swjtu.edu.cn

[1] School of Mechanical Engineering, Southwest Jiaotong University, Chengdu 610031, China

[2] Department of Design, Northumbria University, Newcastle upon Tyne NE1 8ST, UK

Keywords JSP scheduling · Artificial intelligence · Smart factory · Smart distributed scheduling

Introduction

Job shop scheduling or the job-shop problem (JSP) is an optimization problem in which various manufacturing jobs are assigned to machines at particular times while trying to minimize the makespan. Scheduling has direct impacts on the production efficiency and costs of a manufacturing system, thus it has attracted a great deal of research attentions since 1956.

However, JSP is usually a NP combinatorial optimization problem. When scaling up a problem, the existing optimization methods concentrated on centralized scheduling or semi-distributed scheduling meet great challenges in terms of computational stability and time. Now under the Industry 4.0 environment, the scheduling should deal with a smart manufacturing system supported by novel and emerging manufacturing technologies such as mass customization, Cyber-Physics Systems (CPS), Big Data, the Internet of Things (IoTs), Artificial intelligence (AI), Digital Twin, and SMAC (Social, Mobile, Analytics, Cloud). The scheduling research needs to shift its focus to smart distributed scheduling modeling and optimization.

In order to shifting traditional scheduling into smart distributed scheduling (SDS), the research issues are (1) what traditional scheduling methods and techniques can be combined and reused in SDS and (2) what are new methods and techniques required for SDS. Therefore, in this paper, we first review existing researches aiming to answer the first question and discusses a future research direction in SDS to reduce the complexity of centralized scheduling and support smart manufacturing systems.

The contributions of this paper are twofold: (1) reviewing the up-to-date JSP models and solution approaches to identify their current usages and challenges for SDS, and (2) exploring the new development directions, identifying new techniques for reframing problems in JSP to support smart factories in the future and constructing a framework on solving the JSP problem under Industry 4.0.

The remainder of the paper is structured as follows: "Research strategy and literature review method" section introduces our literature review method and resultant scheduling models found in our literature review are classified in "Scheduling model classification" section and their corresponding scheduling algorithms are summarized in section. In "Optimization algorithm for scheduling" section, we discuss the identified new techniques to change a centralized scheduling into a smart distributed scheduling and the new development framework for realizing a smart distributed scheduling, which is followed by conclusions in "Conclusion" section.

Research strategy and literature review method

Research strategy

We structure scheduling researches into two aspects: problem modeling and model solving methods. Our research strategy shown in Fig. 1 illustrates our research roadmap.

First, we classify JSP problem into different types, reclassify job shop scheduling models by the problem spaces (or structures) to form several JSP structural models and then analyze the features of the traditional structural models and explore characteristics of JSP structural models with smart factory under Industry 4.0.

Secondly, the algorithms of solving the scheduling problem are reviewed and analyzed. According to the method classification, the related work and applications are reviewed, including early work (mainly single algorithm) and recent work (mainly involving various combinations of algorithms and some new algorithms). The advantages and disadvantages of algorithms to the traditional scheduling are concluded and then the adaptability and challenges of the algorithms to be used in the smart distributed scheduling are summarized.

Thirdly, driving forces in Industry 4.0 for smart distributed scheduling are studied, including IoT, CPS, smart factory, cloud computing, big data, deep learning, self-decision and other factors. The framework on solving the JSP problem under Industry 4.0 is then constructed with key enabling technologies. The implementation steps of a distributed scheduling algorithm are discussed under Industry 4.0.

Finally, according to the above discussion, the paper summarizes the present solving method of the JSP problem and the future research trend in Industry 4.0.

Literature review method

Since the first mathematics model for scheduling with two machines was built by Johnson (1954), scheduling has been

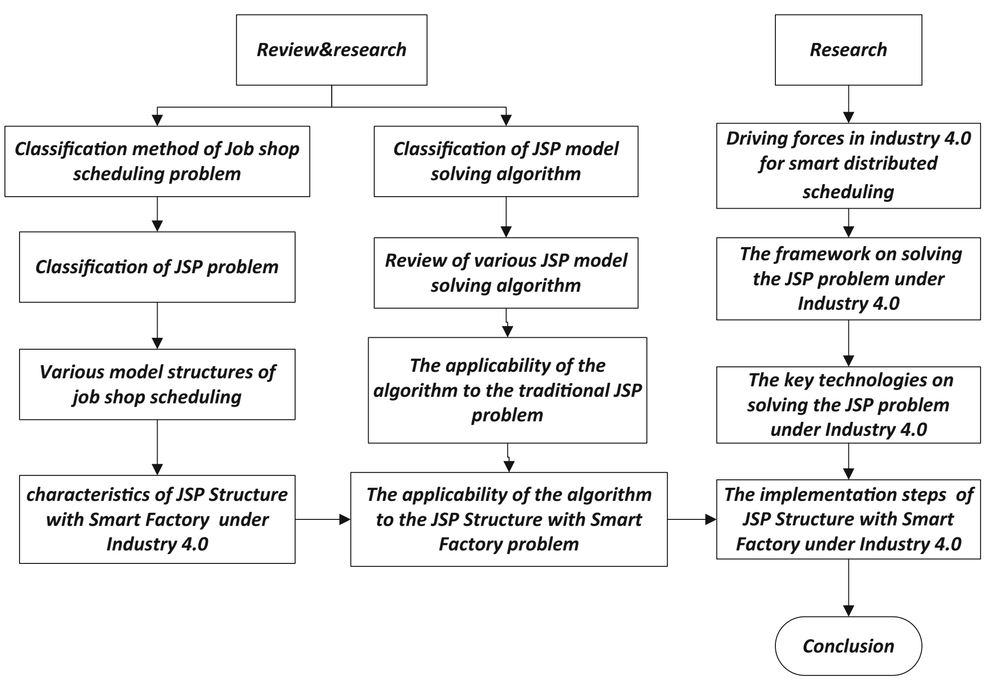

Fig. 1 Research strategy

a hot research topic in manufacturing with extensive research and literatures. JSP is a NP complete problem when the number of machines is more than 2 (Garey et al. 1976). It can be defined as a problem that a given set of jobs J_i ($i = 1, 2, \ldots, n$) need to be scheduled on a set of machines M_j ($j = 1, 2, \ldots, m$) in a way to minimize the makespan (Geyik and Cedimoglu 2004; Çaliş and Bulkan 2015). When assigning one job on one machine, it must meet some constraints. Firstly, each job assigned on a machine is associated with a given order and a machining (or performing) time. Secondly, each machine can perform only one job at any moment (Chen et al. 2012). Lastly, the performing (machining) time of a job is fixed, and once the job is started, it cannot be interrupted (Ju 2007). For an advanced planning and scheduling (APS) (Van Eck 2003), scheduling techniques consider a wide range of constraints to produce an optimized solution, including material availability, machine and labor capacity, due dates, inventory safety stock levels, cost, distribution requirements, sequencing for set-up efficiency (Lin et al. 2012). All of above inputs and constrains can be regarded as the general constrains for APS. The minimum makespan should then be achieved with utilizing scheduling techniques and the general constrains.

For this review, we searched the Google scholar from 1986 to 2016 with key words "job shop scheduling". We found several early review literatures on scheduling. Graves (1981) focused on production scheduling, while Jain and Meeran (1998) paid attention mainly to job-shop scheduling techniques and Akyol and Bayhan (2007) focused on the evolution of production scheduling with neural networks. Recently, Neto and GodinhoFilho (2013) focused on the applications of Ant Colony Optimization approach and Floudas and Lin (2005) themed on AI solution strategies in JSP. These reviews are mainly focused on various JSP optimization techniques, while this paper is focused on reframing the job shop scheduling problems under the new Industry 4.0 environment. Therefore, after filtering some literatures in the early review papers, this review only selected 122 papers in our discussion mainly on scheduling model classification (or definition) and optimization algorithms for scheduling. They are detailed in "Scheduling model classification" and "Optimization algorithm for scheduling" sections.

Scheduling model classification

We classify scheduling models in order to see their features and limitations for Industry 4.0 environment.

Scheduling involves determining the allocation of plant resources. For an earlier and more extensive explanation of the diverse aspects of scheduling models, there are some direct reviews such as Graves (1981), Floudas and Lin (2004) and Floudas and Lin (2005). There are also various classification methods for plant scheduling. Scheduling can be classified by production sources of demand, the number of machine tools, complexity of a production system, performance index, characteristics of a production environment, processing characteristics of the operation and resource constraints (Lin et al. 2012; Ju 2007; Graves 1981).

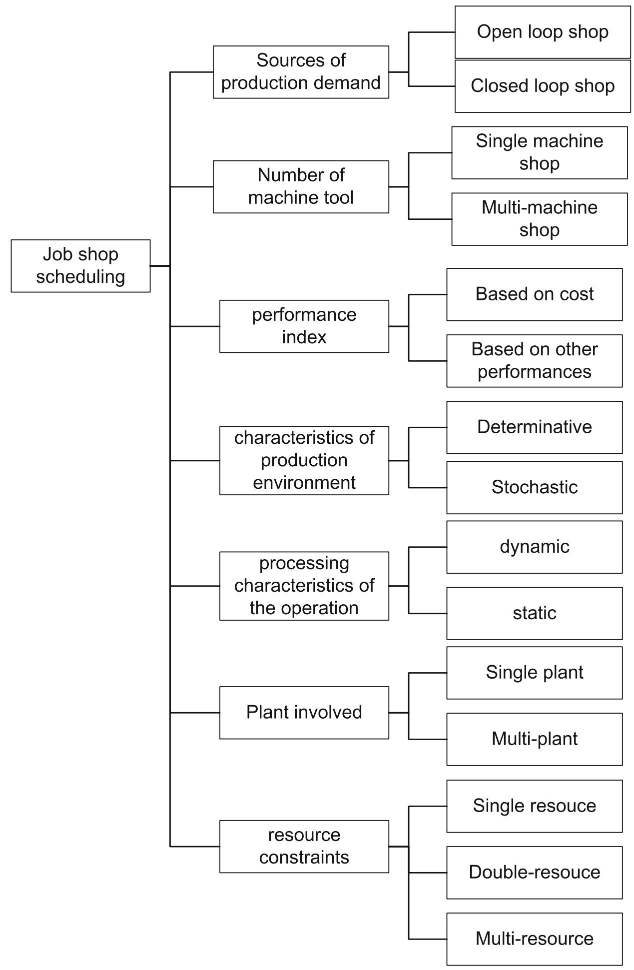

Fig. 2 Types of job shop scheduling

Based on the literature review and considering the plants involved (Lin et al. 2012), there are 16 job shop scheduling models (see the right hand side of Fig. 2), which can be classified by sources of production demand, number of machine tools, performance index, characteristics of production environments, processing characteristics of operations, plant involved and resource constraints. The specific classification is summarized and shown in Fig. 2.

Adding more characteristics and constraints in the problem, its problem space (or structure) will become more complex. For reframing the scheduling problem and broadly understanding what they are and what are their key features, we re-classify the job shop scheduling models by the problem spaces (or structures) into five types of structures, namely basic type, multi-machine type, multi-resource type, multi-plant type and smart factory type. The five structures are shown in Table 1.

Basic type (JSP) Although the basic type of job shop scheduling model (JSP) is the simplest model among five types, most of basic JSPs are still NP-hard problems. In this kind of model, a specified operation is processed on a specified machine tool and no more machines can be chosen. Optimization algorithms or multi-objective optimization algorithms are employed to make the makespan or cost or both of them minimum and achieve a final optimal operation sequence.

Flexible JSP Flexible job shop scheduling model (FJSP) is advanced and complex JSP as machines can be selected for some or all operations. If all of machines can be chosen in the operations, it is called Complete FJSP (C-FJSP). If only some of machines can be chosen in the operations, it is called Part FJSP (P-FJSP). Using multi-objective optimization algorithms can generally achieve both the minimum makespan and the minimum cost with balanced workloads on machines. After that, operation sequences with selected machines can be obtained. Most scheduling algorithms focus on JSP and FJSP nowadays, although their structures are far away from an actual manufacturing system. Therefore another two kinds of extensive JSP are introduced.

Multi-resources FJSP model (MrFJSP) Fora FJSP model, when multi-resources are considered, it is called FJSP model with Multi-resources or Multi-resources FJSP model (MrFJSP). Production capacity of a job shop is restricted by machines, tools, dies, fixtures, operators, vehicles, robots and other manufacturing resource constraints. A machine might be available for some operations, but for scheduling it is still subjected to other available resources as constraints. The scheduling has to assign different available resources at a time including machines and other resources to a process, which is more sophisticated and dynamic than FJSP obviously. Besides the inputs and constraints of FJSP, plant resource information and layout information are the inputs and constrains of the model. The purpose of scheduling is to generate operation sequences of jobs on each machine to achieve the optimization of certain parameters under the constraint conditions. Usually, multi-objective optimization algorithms are applied to achieve the objectives of FJSP with minimum resource transition times. This type of models can be regarded as extensions to FJSP and JSP mathematical model. Some scholars such as Lin et al. (2012), used a disjunctive graph to build a unified model, and other scholars (Ju 2007) used a general analytical mathematical model.

Multi-plants-based MrFJSP (or MpFJSP) Based on a MrFJSP model, when multi-plants and transportations among them are taken into account, it becomes MpFJSP model or MrFJSP model with Multi-plants and transportation (MpFJSP). This kind of model is the most complicated and dynamic model in nowadays scheduling models. Due to different resource management models and dynamic reschedules, it is difficult to obtain an optimal solution for a centralized scheduling. Moreover, it must be most flexible and adaptable because any abnormal change and disturbance could influence all of the other plants. So this type of model can be regarded as a semi-distributed scheduling model. This is to say, it is almost a centralized scheduling model except when the rescheduling is employed to meet the dynamic needs among plants, it turns into a distributed scheduling model. Agents are always used in a multi-plant environment to deal with collaboration scheduling between plants. In one plant, it is the same as a multi-resource environment, otherwise, the inputs and constraints in MrFJSP also involve multi-plant supply chains, pickups and deliveries. Multi-objective optimization algorithms are devoted to find a solution for the minimum makespan, cost, tardiness and mileage of vehicles. An operation sequence with selected machines and resources cross plants is then acquired.

All above four models are concentrated on centralized scheduling or semi-distributed scheduling. These types of scheduling are difficult in a speedy way to response to real-time and dynamic changes to concerned elements in a distributed manufacturing system. For a NP-hard problem, if real-time dynamic changes are occurred simultaneously, a rescheduling should cost more time and often needs a longer recovery period with severe disruptions to a manufacturing process. Actually, these changes can be happened frequently such as emergency events, new orders, cancelled orders and machine failures at any time. Therefore, it is an essential requirement for scheduling in response to mass customization and adaptive manufacturing. But it is impossible that the whole system is interrupted to schedule repeatedly and continually even in one plant. Therefore, smart factory with smart distributed scheduling is demanded in the future because smart agents with self-organization, self-study and self-decision-making features can schedule their own processes indeed. This type of scheduling is named as *MpFJSP with Smart Factory* (*or SFFJSP*) in Table 1. This will be discussed in Sect. 5 with more details.

Optimization algorithm for scheduling

After the identification of a new type of scheduling for Industry 4.0, we continued to summarize the characteristics of existing optimization algorithms and identify their challenges for SFFJSP. In the Table 1, the optimization algorithms for scheduling are divided into two kinds, which are mono/multi-objective algorithms and distributed optimiza-

Table 1 Main structures of JSP

Structures	Characteristics	Objective	Constraint	Algorithm	Target
Basic type: job shop scheduling model (JSP)	Concentration; simplest in scheduling problem but a NP problem under mostly condition	Min makespan (Min Cost)	The general constrains for APS	(Multi-objective) optimization algorithm	Operation sequence
Multi-machine type: flexible job shop scheduling model (FJSP)	Concentration	Min makespan	The constraints of JSP	Multi-objective optimization algorithm	Operation sequence with selected machine
	Usually NP problem	Balancing workloads (Min Cost)	Machines can be selected for some or all operations		
Multi-resource type: FJSP model with multi-resources (MrFJSP)	Concentration	Min makespan	The constraints of FJSP	Multi-objective optimization algorithm	Operation sequence with selected machine and resource
	Usually NP problem	Balancing workloads	Resource information		
	Dynamic	Resource transition times (Min Cost)	Plant layout information		
Multi-plant type: MrFJSP model with multi-plants and transportation (MpFJSP)	Semi-distribution	Min makespan	The constraints of MrFJSP	Multi-objective optimization algorithm	Operation sequence with selected machine, resource and plant
	Usually NP problem	Min tardiness	Multi-plant chain		
	Higher dynamic	Min mileage of vehicles (Min Cost)	Pickup and delivery		
Smart factory type: MpFJSP with smart factory (SFFJSP)	Distribution	Job objective	The constrains of MpFJSP	Distributed optimization algorithm	Operation sequence with selected machine, resource and plant
	Highest dynamic	Resource objective	System rules		
	Real-time	Plant objective	Real-time information		
	Self-organization	System objective (machine utilization rate monthly production efficiency monthly)			
	Highest flexible				

tion algorithms. For the traditional problems, JSP, FJSP, MrFJSP and MpFJSP are concentrated on semi-distributed scheduling problems, which can be solved by mono/multi-objective optimization algorithms. For the future job shop scheduling problem under SFFJSP, it is a smart distributed scheduling problem, which should use distributed optimization algorithm to deal with. In fact, in a smart distributed scheduling problem, the system is divided into several local subsystems and every subsystem builds its own structure according to the related smart agent(s). That is to say, the original problem can be decomposed into different smaller and more flexible parallel sub-problems and all these sub-problems can be dealt with separately, therefore we can use the mono/multi-objective optimization algorithms to solve sub-problems more easily than a concentrated scheduling problem and achieve better solutions with less time.

There are many ways to solve a traditional job shop scheduling problem and many scholars have also made a summary of this work, such as the early scholar, Jain and Meeran (1998), who classified, introduced and compared various earlier algorithms. In recent years, due to the development of intelligent algorithms, most of the scholars (such as Çaliş and Bulkan 2015) pay more attention to intelligent algorithms, meta-heuristic and some special forms of JSP.

The optimization algorithms for scheduling are mainly divided into exact optimization methods and approximate

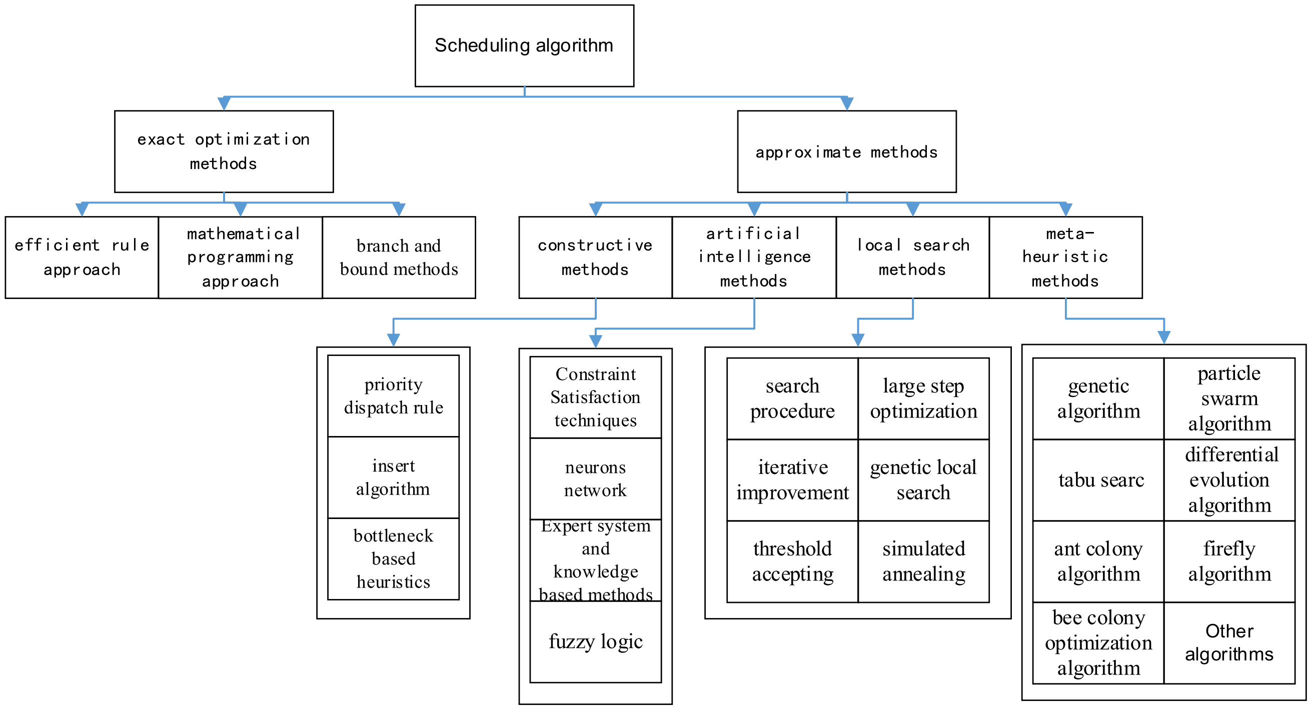

Fig. 3 Optimization algorithms for scheduling

methods. The exact optimization methods include efficient rule approaches, mathematical programming approaches, branch definition methods, and etc. The approximate methods include constructive methods, artificial intelligence, local search and meta-heuristic algorithms. In the smart factories oriented scheduling under Industrial 4.0, smart agents or intelligent bodies become the dominant factors, so that a previous centralized scheduling system can be replaced with multiple connected smart scheduling agents. For a single agent, the complexity of scheduling drops substantially. Therefore, it is very possible that the earlier methods only for small scale scheduling can be used again in specific circumstances. This is the reason why we still have a summary of the various methods including contemporary and early approaches in order to give play to the advantages of different methods in the future smart manufacturing system. Each method is subdivided with algorithms, as shown in Fig. 3.

Exact optimization procedure methods

Mathematical programming and operational research are applied to achieve the global optimal solution or deterministic optimal solution from 1950s to 1980s.

(1) Early studies of *exact optimization procedure methods*

Efficient rule-based methods are the earliest approaches in this field. According to the input data, a series of preliminary rules can be established, which determine the processing order exactly, these methods can obtain an exact optimum solution. Johnson (1954) proposed a set of rules called Johnson rules to solve a two-machine flow job shop problem with a determined order. The criterion of minimizing the makespan was set up in this paper, which has brought a great influence on later JSP researches. Wagner (1959) found mathematical programming techniques could solve JSP problem optimally. However, researchers also found mathematical programming techniques had their own shortcoming because of the excessive computing time required or resulting in poor quality solutions. Therefore it turned to enumerative methods, especially branch and bound methods (B&B). Brooks and White (1965) and Lomnicki (1965) applied B&B for the exact optimal solution of JSP. Hefetz and Adiri (1982) discussed an efficient optimal approach for the two-machine JSP in which each operation is carried out with a unit processing time. All above problems are particular problems and can be done in polynomial time with efficient rule-based methods.

(2) Combination algorithms based on exact optimization procedure methods

Manne (1960) mixed discrete integer and linear programming (MIP) approaches and proposed a common form of mathematical formulation, which included a linear objective function, a series of linear constraints and the binary integer variables to decision. This formulation involved considerably fewer variables and computed more efficiently than Wagner's.

Because the B&B method can calculate the lower bound of the some subsets, it is an effective method to solve the scheduling problem with a better solution. McMahon and Florian (1975) presented a successful application in which branching was built by identifying the critical job with the maximum lateness and then determining all the other jobs with longer due dates. Sarin et al. (1988) and Potts and Wassenhove (1985) improved B&B methods separately. The improved methods are different on the surface, but both of methods focus on analysis rules, bounding mechanism and the generation of upper bound.

The exact optimization methods, such as efficient rule-based approaches, mathematical programming approaches and branch definition methods, can achieve the exact optimum solution in polynomial time for specified JSP problems. But only small scale problems can be solved with the exact optimization methods. As the best exact optimization methods for more complex problems, the B&B methods can obtain the optimal solution in theory, but it is difficult to have real practical applications because of its complexity. Moreover, for larger scale problems, B&B methods take too much time. For a nm JSP problem, there are $(n!)^m$ possible solutions. So for a large-scale problem, the exact optimization methods are not possible to complete calculations in a good responding time. After decentralizing scheduling, each smart agent will face very different scales of scheduling problems. Using this method, we can get the exact solution for a small scale problem.

Approximate methods

With the continuous development of computer technology and intelligent algorithms, the research methods related to JSPs have gradually changed from exact optimization procedure methods into the approximation methods since 1980s.

Constructive methods

Constructive methods can find the JSP solution fast. They include three typical methods: priority dispatch rule, insert algorithm and bottleneck based heuristics.

(1) Early studies of the priority dispatch rules method

The first approximation procedure of JSP is the priority dispatch rules method, it uses the priority dispatch rules such as the shortest processing time, the longest remaining total processing time, the earliest delivery time and the selection of the same machine on the first working process. This method is always easy to process with dramatically reduced computation (Barker and McMahon 1985; French 1982; Morton and Pentico 1993). All the operations are dispatched based on their priorities and the operation with the highest priority is selected to be scheduled at first. So the key technology focuses on selecting the best priority rules according to different actual problems. For example, if reducing average flow time of all jobs is the most important, we can choose the shortest processing time rules. But if optimizing the maximum delay is the most important, we should turn to the earliest delivery time rules. Usually several priority dispatch rules are built simultaneously to achieve a satisfied solution.

(2) Combination algorithms and recent studies based on the priority dispatch rules method

Ingimundardottir and Runarsson (2011) introduced the learned linear priority dispatching rules for JSP in which linear classification was used for dispatching rule framework to identify good choices from inferior ones by a supervised learning approach and experimental studies showed that the output is better than that of using common priority dispatching rules. Zahmani et al. (2015) built a simulation model for makespan optimization, used different dispatching rules for each machine to select the best rule for every new scheduling problem and showed advantages of using multiple priority dispatching finally. Paul et al. (2016) adopted preference selection index method for ranking priority dispatching rules for scheduling an assembly job shop.

(3) Early studies of insert algorithms

Insert algorithm was developed by Rosenkrantz et al. (1977) for dealing with the travelling salesman problem. Inserting operations or jobs into partial schedules one by one usually could outperforms priority rules. Nawaz et al. (1983) used insert algorithm to handle the permutation flow shop problems.

(4) Combination algorithms and recent studies based on insert algorithms

By changing the rule for selecting an element to the next insertion, especially combining the insert algorithm with beam search or considering different insertion orders, Werner and Winkler (1995) and Sotskov et al. (1999) applied the algorithm for dealing with JSP and obtained a better result. Lian and Mesghouni (2014) put forward an improved inserting algorithm (IA), in which firstly a pre-schedule was obtained through heuristic algorithm and then maintenance tasks were inserted into the pre-schedule scheme to realize the dynamic scheduling.

(5) Early studies of bottleneck based heuristics methods

Bottleneck based heuristics methods, such as Shifting Bottleneck Process and Bean Search, are more sophisticated approaches to balance good results and time consuming. Shifting Bottleneck Process (SBP) was applied first by Adams et al. (1988), in which the original problem was relaxed and decomposed for the sub-problems of single machine scheduling and was solved separately later. One bottle machine was chosen in each round of iterations and the process order of all the jobs on the bottle machine was fixed, so the process could be repeated until whole machine orders were fixed. Dauzere-Peres and Lasserre (1993) modified this procedure by considering delay precedence constrains (DPC) of sub-problems. Balas and Vazacopoulos (1998) presented that B&B could solve DPC of sub-problems.

(6) Combination algorithms and recent studies based on bottleneck based heuristics methods

Wenqi and Aihua (2004) proposed an improved SBP algorithm (IBP) for the JSP and proved IBP could guarantee feasible solutions. By studying structural properties of an extended disjunctive graph model, Liu and Kozan (2012) developed a hybrid shifting bottleneck procedure algorithm to treat the parallel-machine job-shop scheduling problem.

Constructive methods, such as priority dispatch rule, insert algorithms and bottleneck based heuristics, can acquire a JSP solution very quickly sometimes, but infeasible solutions may be generated especially when the problem is sophisticated. In order to improve the quality of solutions, it is usually necessary to set up complex heuristic rules. For a sophisticated system, there are so many rules that are restricted each other and even trapped in a loop or contradictory at times. Thus it is difficult to find a feasible solution to meet all rules. However, the shorter the time of acquiring solution is, the faster the dynamic scheduling responses, this is attractive to the SSFJSP.

Artificial intelligence methods

In the summer of 1956, a group of outstanding young scientists jointed together, discussed on a series of related problems of machine simulating intelligence and proposed the term "artificial intelligence" at first time. It represents a milestone of the official birth of the discipline of "artificial intelligence" (AI). AI is a uniform name concerned with the field of computer science dedicated to the development of programs that attempt to replicate human intelligence (Fonseca and Navaresse 2002).

(1) Early studies of artificial intelligence methods

Constraint satisfaction techniques aim at exploring and reducing the effective size of the search space by applying constraints to determine the whole order and sequence by selecting variables and allocating possible values, which are referred to as variable and value ordering heuristics. Although belonged to the domain of AI, many Constraint Satisfaction methods for JSP apply a systematic tree search and are accompanied by B&B algorithms (Jain and Meeran 1998).

Simon and Takefuji (1988) presented a two-dimensional Hopfield TSP type matrix of neurons and encoding strategies to solve JSP for minimizing the sum of all the starting times of each job's last operation. Akyol and Bayhan (2007) provided an extensive literature review on the applications of Neural Networks (NNs) to scheduling problems and divided them into four categories: Hopfield type networks (HNN), multilayer perceptrons, competition based networks and hybrid approaches in accordance with the different structures.

Many literatures focused on this HNN to solve a static scheduling problem, while Fnaiech et al. (2012) applied it to solve joint production and maintenance scheduling problems. However it is difficult to be used in actual JSP because the variables involved are so large that various problems have brought about, i.e., the computational efficiency is low, and it may not converge to good quality solutions.

A multilayer feed forward network, called back-propagation (BP) network, was put forward and got a wide range of applications. BP networks are not directly involved in optimization, and it uses a training data set to its input and output layers and trains itself by back-propagation algorithm. After training, application of the network involves only the computations of the feed forward phase (Fausett 1994). The performances of these networks are generally decided by their generalization capabilities and generalization accuracy.

A common competitive ANN is composed of an input layer and a competition layer of processing nodes. Each node on the input layer is connected to every node of the competition layer with the established connection weights. The sum of input value of a node in the competition layer competes with those of neighbor nodes (Kartam and Tongthong 1998), and the most extreme one called Winner Take All will have a nonzero output signal. Some researchers study competitive networks to solve scheduling problems by optimizing and classifying problems more efficiently and simply. Min and Yih (2003) applied this network to train with the Kohonen learning rule, and developed a multi-objective scheduler to select dispatching rules for initial variables to satisfy the objective finally.

Expert system and knowledge-based methods are composed of knowledge base and reasoning mechanism. The knowledge base includes a number of rules, processes, and heuristics information. The reasoning mechanism is used to select a strategy to deal with the knowledge in the knowledge base. The more famous expert systems are: ISIS, MPECD, OPIS, SONIA (Ju 2007) and so on. Because of the limited capacity of a single expert system, some scholars have put forward the parallel or distributed strategies to solve the scheduling problem. Van Dyke Parunak (1985) presented a dynamic research of flexible manufacturing system which is carried out by using the distributed decision making method of multi-agent structure. Chen et al. (1998) studied scheduling problem of production line by multi-agents.

(2) Combination algorithms and recent studies based on artificial intelligence methods

A series of improved constraint satisfaction techniques have been presented with different objectives, such as Pesch and Tetzlaff (1996), Sadeh and Fox (1996). Later, these methods were applied successfully in scheduling sequence, planning process and planning vehicle routes. More detailed information about constraint satisfaction techniques in different fields can be found in Apt (2003) and Rossi et al. (2006). Barták et al. (2010) gave an overview of constraint satisfaction techniques in planning and scheduling.

In recent years, among artificial intelligence methods, only NNs have been developed. NNs have been combined with other methods to form hybrid approaches to overcome some of its limitations. In hybrid approaches, one of the combined methods acts as the main problem solver meanwhile the other method assists it. EANNs can be considered as the combinations of ANNs and evolutionary search procedures and the two approaches hybrid their functions together to deal with the problem more efficiently. Different from other hybrid approaches, evolution is united by Artificial Neural Networks usually with 3 factors: the connection weights, the structures and the learning rules. Adibi et al. (2010) put forward a hybrid method based on variable neighborhood search (VNS) and artificial neural network (ANN) for dynamic job shop scheduling to deal with random job arrivals and machine breakdowns. Xanthopoulos and Koulouriotis (2015) applied BP neural networks to approximate the functional relationship between dynamic sequencing priority rules and performance metrics of the production system. The results of the trained BP neural networks for scheduling can be used to predict outputs of dispatching rule systems, direct to build new dispatching heuristic and significantly decrease the time of simulation studies. Wang and Jiang (2016) proposed general regression neural network (GRNN) to establish the explicit mapping function from the data points in high-dimensional space to the data points in low dimensional embedded space based on locally linear learning, then least square-support vector machine (LS-SVM) was trained and acted as a solver to select an appropriate rescheduling method.

Constraint Satisfaction techniques, NNs and Expert system and knowledge based methods are representatives of artificial intelligence methods. All of these methods have been employed widely, such as BP network due to its nonlinear mapping ability, self-learning ability, fault to learn and predictive capability. But it is also well known BP network exist a lot of shortcomings, such as dissatisfied training approximation and generalization, over fitting or over study, non-convergence, structure choice without scientific rules. Some of expert system and knowledge based methods are so-called “distributed”, but it makes decision by the system not each agent itself. Building the rules of whole system is difficult as there are too many factors should be considered for a large and complex system. However, it is possible that the trained approximation and generalization can be improved with Deep Neural Network, which can be useful for the real-time dynamic and indeed distributed scheduling in the future.

Local search methods

Local search methods (Aarts and Lenstra 1997) usually consist of a finite set of solutions, an optimized function or a set of optimized functions and a searching strategy. According to the searching strategies, typical local search methods include greedy randomized adaptive search procedure (GRASP), iterative improvement (IM), threshold accepting (TA), large step optimization (LSO), genetic local search (GLS) and simulated annealing (SA) (Jain and Meeran 1998). Among these methods, SA becomes the most popular local search algorithm to deal with the FJSP currently.

(1) Early studies of SA

Simulated annealing first presented by Kirkpatrick et al. (1983) is a random-oriented local search method and also a meta-heuristics method. It is similar to a statistical physics process of a heated solid’s annealing from its maximum energy state to minimum state gradually by controlling parameter. Due to its high time cost, many papers are devoted to reduce searching efficiency by being mixed with different meta-heuristic methods.

(2) Combination algorithms and recent studies based on SA

Xia and Wu (2005) used SA to avoid being trapped in a local optimum and particle swarm optimization method to enhance high search efficiency and the solutions showed it was a practical method for a multi-objective flexible job shop scheduling problems at a large scale. In more recent study, Zorin and Kostenko (2014) applied SA and a multi-layer model to plan scheduling without the exact time of job beginning and ending by estimating the time of execution and proved asymptotic convergence of the algorithm. Shivasankaran et al. (2015) proposed a mixed method of SA and immune algorithm to solve sorting limits. Harmanani and Ghosn (2016) proposed an implemented SA algorithm to deal with the nonpreemptive open JSP problem by efficiently exploring the solution space. Zandieh et al. (2017) proposed an improved imperialist competitive algorithm (ICA) for the FJSP scheduling problem, enhancing the performance with a hybridization of ICA with SA. In order to use manufacturing resources of job shops more effectively in satisfying customers, Güçdemir and Selim (2017) proposed a simulated annealing based simulation optimization approach.

Jain and Meeran (1998) compared local search methods and gave a conclusion that although the best solution could be searched, all of local search methods cost too much time. That is to say, all of these kinds of methods, including GRASP, IM, TA, LSO, GLS and SA, can achieve optimum solution if enough time is given and this is the greatest advantage undoubtedly. However, with the mass customization and global manufacturing, job shop scheduling should be arranged as fast as possible. Therefore these methods are rarely used independently in recent years and are combined with meta-heuristics methods usually.

Meta-heuristic methods

Reeves (1993) proposed a heuristic algorithm to find an optimal solution with a complex model. One of heuristic algorithms called meta-heuristic algorithm with random number search techniques are used in a very wide range of practical problems. Early generic meta-heuristic algorithms include genetic algorithm and Tabu search method (Jones et al. 2002). By far, a lot of other algorithms, such as the ant colony algorithm, particle swarm algorithm, differential evolution algorithm, firefly algorithm etc., all of which are the imitation methods of nature or biological circles, being applied in the job shop scheduling problem (JSP) and achieved good results.

(1) Early studies of genetic algorithm for JSP

Genetic algorithm (GA) is one of popular meta-heuristics which is based on the genetic evolution mechanism of biology. One of GAs' main characteristics is to directly operate on the problem structure without derivation and function continuity limitation. GAs also have the inherent implicit parallelism and global searching ability and can adjust search directions automatically and self-adaptively. The original GA was used to JSP by Davis (1985) who formed a preferred sequence of operations for every machine in which GA is an indirect method. After that various efforts have been made to adapt genetic algorithms to solve different JSPs and have been improving the performance of genetic search by integrating other heuristic methods. Falkenauer and Bouffouixm (1991) enhanced this method by encoding all of operations of each machine as a preferred string of symbols.

(2) Combination algorithms and recent studies based on GA

Cheng et al. (1999) and Gen and Lin (2014) reviewed the studies on solving JSP problems by GAs. Kuczapski et al. (2015) studied the GA and proposed a method to generate the good initial population by evolving priority dispatching rules to the arrival of optimal final solutions. Jalilvand-Nejad and Fattahi (2015) put forward an incorporated integer linear programming model for cyclic FJSP and compared GA and SA showing that the former is more efficient than the latter. Zhang and Chong (2016) proposed a multi-objective genetic algorithm incorporated with two problem-specific local improvement strategies to solve a bi-objective optimization problem. Zhang et al. (2017) took into account the shortest processing time and the balanced use of machines, and put forward the multi-population genetic algorithm based on the multi-objective scheduling of flexible job-shop.

(3) Early studies of Tabu search for JSP

Tabu search (TS) is a global iterative optimization technique and also one of hot meta-heuristics which applies smart search and store history memory to gain global optimum rather than being trapped at local optimum. The earliest study of TS was done by Glover (1986). Although TS is a simple search procedure, it can prohibit the moves which is the same as or similar to the previously achieved solutions according to the search information stored in the memory, and then avoid local optimum solutions.

(4) Combination algorithms and recent studies based on TS

The summary and comparison of this method used in JSP can be found in the book of Glover and Laguna (2013). Meeran and Morshed (2012) devoted their efforts to combine GA with TS to handle JSP by highlighting the advantage of global parallel search of the former approach and the advantage of local optimum avoidance of the latter approach. Peng et al. (2015) integrated a TS process with path relinking (PR) to obtain better solutions to solve JSP by building a path which could connect the initiating solutions to optimized solutions and select competitive solutions more effectively and better. They also used it for an unsolved problem remained for more than 20 years. Li and Gao (2016) proposed an effective hybrid algorithm which hybridizes the GA and TS for the FJSP with the objective to minimize the makespan.

(5) Early studies of ant colony optimization for JSP

Ant colony optimization (ACO) which imitates the process of ant colony foraging is another meta-heuristic presented by Colorni et al. (1991), who also is the first researcher using this approach to solve the NP-Hard Traveling Salesman Problem. Bullnheimer et al. (1999) applied this approach to consider the vehicle routing problems. Merkle et al. (2002), Blum and Sampels (2004) employed ACO to solve different scheduling problem. Using ACO-based algorithms to solve scheduling problems and other problems was reported by Dorigo (1992).

(6) Combination algorithms and recent studies based on ACO

Some researchers improve optimized procedures by integrating other algorithms with ACO to achieve the better quality of solution or better efficiency, such as taboo search (Huang and Liao 2008), beam search (Blum 2005), knowledge-based (Xing et al. 2010), immunity algorithm (Xue et al. 2015), two-generation pareto (Zhao et al. 2015). As ant colony algorithm is a kind of self-organized parallel algorithm with positive feedback, in recent years, some researchers devoted to deal with dynamic JSP by ACO (Saidi-Mehrabad et al. 2015; Huang et al. 2013). Neto and GodinhoFilho (2013) overviewed ACO-based applications in scheduling and provided a perspective prospect of the future trend. Wang et al. (2017) proposed an improved ACO algorithm. The main improvements include selecting machine rules, initializing uniform distributed mechanism for ants, changing pheromones guiding mechanism, selecting node method, and updating pheromones mechanism. Huang and Yu (2017) proposed an effective ACO algorithm. Five enhancements are made in the proposed algorithms including: a new type of pheromone and greedy heuristic function; three new functions of state transition rules; a nimble local search algorithm for the improvement of solution quality; mutation mechanism for divisive searching; and a particle swarm optimization (PSO)-based algorithm for adaptive tuning of parameters.

(7) Early studies of particle swarm optimization for JSP

Particle swarm optimization (PSO) stemmed from the behavior of birds' prey is a computational evolution meta-heuristic technique, which was originally put forward by Kennedy and Eberhart (1995). Based on the observation of the regularity of the prey activities of flying birds, a model with swarm intelligence is built and improved by sharing individual information in the population so as to obtain the optimal solution. Compared with the genetic algorithm, due to no crossover and mutation operations and few parameters need to be adjusted, the advantage of PSO is that it is easy to implement and practice. The inherent drawback of PSO is lacking of global convergence owing to large reduction of velocity values.

(8) Combination algorithms and recent studies based on PSO

Xia and Wu (2005) combined PSO which assigns operations on machines with SA which schedules operations on every machine to solve FJSP hierarchically. In order to overcome the drawback of PSO, researchers recently explored combinational methods of PSO with others to solve JSP and FJSP, such as Baykasoğlu et al. (2014), Yin et al. (2015), Nouiri et al. (2015), Teekeng et al. (2016), Huang et al. (2016). Recently, Singh and Mahapatra (2016) used an operator in genetic algorithm into mutation operation and logistic mapping to generate chaotic numbers rather than random numbers. Chaotic numbers generally mean random and pseudorandom numbers with good statistical properties. Compared with several popular algorithms, this method is more effective on reducing makespan. Muthiah et al. (2016) proposed the hybridization of the Artificial Bee Colony (ABC) and Particle Swarm Optimization (PSO) techniques to minimize the makespan of the shops. In the same vein, Nouiri et al. (2017) proposed a two-stage particle swarm optimization (2S-PSO) to solve the problem assuming that there is only one breakdown.

(9) Early studies of differential evolution for JSP

Differential evolution (DE) was proposed by Storn and Price (1995) and is an evolutionary meta-heuristic technique by imitating the evolution organisms and repeating iteration to reserve the individuals which have adapted to the environment. Compared with GA, this approach can be realized more easily and converged more efficiently for continuous optimization.

(10) Combination algorithms and recent studies based on DE

Ponsich et al. (2009) showed that DE alone could not achieve solution as well as GA or TS and the reason may be that DE is lack of integrity and self-adaptiveness on the permutation representation approach and the mutation operator to a discrete problem or a JSP, although it is practically well for a continuous optimization problem. An improved approach focused on hybridizing a neighborhood search approach with DE to make the local search more efficiently. Ponsich and Coello (2013) combined differential evolution and Tabu search approach to solve the JSP problems and showed this DE/TS algorithm was comparable with the other current advanced techniques. The optimum solution also indicated a common high efficacy of this algorithm by a lot of examples, especially for most of the median sized JSP problems, and it could seek the solution with a satisfactory repeatability. Zhao et al. (2016) got a balance between the global exploration and local search space efficiency by embedding a speed-up neighborhood search procedure for seeking key paths into differential evolution algorithm to solve FJSP.

Zhang et al. (2016) proposed a chaotic differential evolution algorithm (CDEA) with makespan minimization criterion. In the CDEA, logistic mapping is used to generate chaotic numbers for the initialization because it is helpful to diversify the CDEA population and to improve its performance in preventing premature convergence to local minima.

(11) Early studies of firefly algorithm for JSP

Firefly Algorithm (FA) is a more recent approach which was original presented by Yang (2008) and comes from the population behavior of fireflies. Łukasik and Żak (2009) proposed a further research on the firefly algorithm for solving continuous optimization problems.

(12) Combination algorithms and recent studies based on FA

To continuous optimization problems and continuous NP-hard problems, it is very effective. However this approach cannot be applied for solving the discrete optimization problems directly because its learning process is based on the real number. To solve the discrete problems, a set of conversion approaches of the continuous functions should be built, for example attractiveness, distance and movement should be changed into discrete functions. Discrete firefly algorithm (DFA) was introduced by Sayadi et al. (2010) to solve the flow shop scheduling problems. Khadwilard et al. (2012) combined DFA with the SPV rules for dealing with the multi-objective hybrid flow shop scheduling problems. Marichelvam et al. (2014) used this algorithm to solve JSP firstly and discussed the different parameters' setting according to their performance. Karthikeyan et al. (2015) developed a hybrided DFA with a local search approach to solve a multi-objective FJSP by using rules for the initial population. For the hybrid methods, discrete firefly algorithm usually focuses on an extensive search for the solution space while the local search algorithm is generally used to reschedule the results for a speedy and accuracy convergence. Marichelvam and Geetha (2016) propose a hybrid discrete firefly algorithm (HDFA) to solve the FSSPs to minimize the total flow time.

There are some other algorithms to be used for dealing with JSP and the extensive problems, such as fuzzy logic (FL), which allows the imprecise or fuzzy nature of the data in real-world problems (Sakawa and Kubota 2000) and usually introduces especial rules to solve scheduling problems (Canbolat and Gundogar 2004). Bee colony optimization (BCO), which is a population-based search algorithm, introduced by Pham et al. (2005) and first proposed by Chong et al. (2006) for dealing with a job shop problem by the honey bees foraging model and showing a slight quicker than other heuristics approaches sometimes, and so on.

Meta-heuristics algorithms usually come from the meta-heuristic methods used to solve the continuous problems initially. When the method can achieve good results for continuous problems or NP continuous problems, it will be introduced to deal with discrete problems sooner or later. In this process, a set of conversion rules should be built and the continuous functions should be changed into discrete functions. Some meta-heuristic algorithms often win a good efficiency for the global search, but for the local search they are easy to fall into local optimum. However some are just the opposites. So how to integrate these features to generate a new and effective hybrid method to treat JSP and extensive problems is the most popular research.

Characteristics of existing JSPs algorithms and their challenges

According to the above review, the characteristics of existing JSPs algorithms and their challenges for implementing in the SFFJSP in the future are summarized in Table 2.

Smart distributed (or decentralized) scheduling in the future

The traditional JSPs are focused mainly on centralized or semi-centralized manufacturing system. Now under the Industry 4.0 environment, most of elements such as machines are smart or intelligent. So a whole manufacturing system will be smart or autonomous decentralized flexible manufacturing system (Iwamura and Sugimura 2010; Hino and Moriwaki 2002). The JSP scheduling problem will shift its focus to smart distributed scheduling modeling and optimization. The complexity of a centralized big system problem can be decomposed and the highest flexibility can be realized. In this model, smart agents have their own optimized objectives, which can be divided into job objectives, resource objectives, plant objectives and system objectives. According to system objectives of machine utilization rate monthly and production efficiency monthly, a series of rules should be set up. Based on these rules, smart agents plan their scheduling in accordance with real-time information by distributed optimization algorithm. Compared to the traditional concentrated scheduling technology, a satisfied operation sequence with selected machines, resources and plants can be obtained more easily and faster with this pioneering technology of Industry 4.0.

Driving forces in Industry 4.0 for smart distributed scheduling

As we all known, Industry 4.0 leads the fourth industrial revolution nowadays which is the future direction of automation and information technologies in manufacturing field.

Table 2 The characters of optimization algorithms

Optimization algorithms		Advantages for existing JSP	Limitations	Challenges for SFFJSP
Exact optimization procedure methods	Efficient rule methods, mathematical programming techniques and branch and bound methods	Achieve the exact optimum solution in polynomial time for specified JSP problems	Only small scale problems can be resolved	How to decentralize scheduling in smaller scales so as to use these methods to get the exact solution
Approximate methods	Constructive method	Acquired the JSP solution very quickly sometimes	Infeasible solutions may be generated	How to avoid infeasible solutions
	Artificial intelligence method	Especially for dynamic job shop scheduling to deal with random job arrivals and machine breakdowns	Dissatisfied training approximation and generalization and so-called "distributed" scheduling	How to improve generalization and turn into indeed distributed scheduling
	Local Search Methods	Optimum solution can be achieved if enough time is given	Need too much time	How to improve their efficiency
	Meta-heuristic methods	Some have a good efficiency for the global search, and others fall into local optimum	Some are easy to fall into local optimum, and others are with poor efficiency	How to improve or blend methods for get good efficiency and avoid local optimum

With the information communication and exchange between jobs, machines, tools, fixtures, people and other resources, the change from centralized control to decentralized or distributed production processes marks one important character of Industry4.0 (Kagermann 2015).The fourth industrial revolution usually consists of several aspects which are Internet of things, cyber-physical systems, smart factories and cloud computing (Hermann et al. 2016). Each aspect influences the job shop scheduling dramatically. Deep learning is the most popular method of artificial intelligence recent year and should become the major method to solve the fast self-decision of different smart agents.

Internet of things (IoT)

In 1999, the "Automatic Identification Center (Auto-ID)" of Massachusetts Institute of Technology proposed an idea of "all things are connected through the Internet" which is the basic meaning of the Internet of things. With sensor technology and radio-frequency technique, IoT technology can link jobs, machines, tools, fixtures, vehicles, robots and people together to generate "big" data over the whole factory. Computer systems with super power can integrate the network information of personnel, machine, tools and other resources in order to manage and control production process sophisticatedly. It is easy to say that this kind of network control and management with all information of factories by IoT technology can make the structure of job shop scheduling more real and make manufacturing smarter than ever before so as to perform new jobs quickly, meet production demands timely, and improve production process effectively and optimize supply chains in real time.

Cyber-physical systems

Cyber-physical systems (Harrison 2016) can fuse the physical world with the virtual world which integrate the computing simulated process and physical real process. Networks with embedded system, computing and control technologies can control the real production processes in accordance with scheduling optimized by computations, meanwhile virtual scheduling models should be adjusted by physical processes. In this process, the computed scheduling by optimization algorithms is confirmed continuously. At the same time, different from the earlier CPS in which a service or agent should be applied to store and analyze data centrally, current CPS are built by not only RFID tags but also multiple sensors and actuators, network gateways etc., which supports store and analyzes data in a distributed way.

Smart factory

Based on Internet of things and cyber-physical systems, the smart factory could be formed. In smart factories with modular structures, cyber-physical systems surpass IoT by communicating and cooperating with one another and make decentralized decisions possible. When all of the jobs, resources and other things of the whole factory become intelligent and smart, the traditional centralized job shop scheduling is changed to the smart distributed scheduling with decentralized decisions. The difference is obvious because the traditional distributed scheduling usually still needs a centralized scheduling agent. After the system or the scheduling agent computes and optimize uniformly, the computing result is transmitted to the scheduling actuator. The realization of the smart distributed scheduling under Industry 4.0 will be a true wisdom since each smart agent can self-decide its choice and plan the scheduling. Smart agents can remember their machining history, acquire their current states and know their future goals, so they can actively not only sequence but also assign themselves. For example, tools, fixtures and people etc. can arrange themselves for assisting processes while vehicles or robots can master themselves for logistics to the next job. Therefore, the scheduling should be changed into flexible job shop scheduling models with multi-resources, multi-plants, transportation and smart factory (SFFJSP) in the future.

Cloud computing

Integrated with cloud computing, which is one of recent rises of technologies, with the idea of "manufacturing equals services", cloud manufacturing is emerged, which supplies a new structure for job shop scheduling problem. In a cloud manufacturing environment with the CPS, IoT, cloud computing and other advanced technologies, we can establish a sharing and service platform for coordinating regional manufacturing resources and realizing effective sharing and optimal allocations. And then integrating the logistics optimization technology with capability of researching the cooperation mode between multi-plants and logistics enterprises, we can establish a modern cloud manufacturing service platform. With the cloud computing and manufacturing, the structure of scheduling includes multi-plants, multi-suppliers and multi-logistic providers, thus smartly distributed scheduling and decision is the best choice for solving the complex and dynamic problem. At the same time, in the distributed scheduling, each agent makes distributed decisions which can reduce the workload by parallel computing, and the local problem can be solved by the big data and cloud computing technologies.

Deep learning and self-decision

Deep learning applies a complex nonlinear model to represent the relationship between the data, and employs big data to analyze and determine what the end relationship between the data is. With the development of large data, high performance computing and cloud computing technologies, deep learning is approved again, even beyond human wisdom in some ways. In these 2 years, it is becoming the most spectacular area of artificial intelligence. It will be one of the mainstreams in machine learning and be used in more and more industrial fields in the future.

In the field of job shop scheduling, deep learning should bring us a real sense of autonomous learning and independent decisions. For different batch jobs, the scheduling principle is "the first comes, the first serviced". The earlier the jobs arrive, the earlier machining services are scheduled. However for the same batch jobs, preliminary sorting is scheduled by the system at first, which could be random or optimized. And according to this kind of sorting, each subsequent job self-determines whether it adjusts the queue with deep learning technology. It includes two areas. On the one hand, each job judges whether the current sorting can meet its delivery time. If it cannot meet, the job should jump the sequence. On the other hand, on the premise of guaranteeing the delivery time, deep learning technology with autonomous decision-making should be used to forecast efficiency, costs and machine utilization etc. If it could improve efficiency, reduce costs and achieve production line balance after adjusting sequence, then change the priority and adjust the queue. When the information could be transferred between smart jobs, this autonomous decision-making process could be realized.

According to the analysis of large data and real-time condition data collection of machine tools and other resources, by deep learning and improving learning techniques, the system can predict failure and issue the maintenance instructions, and then the jobs involved would respond to choose their own process resources and orders. When the information could be transferred between smart jobs and machines, this self-decision process could be realized.

According to the analysis of large data and real-time condition data collection of machine tools and other resources, with deep learning and improving learning techniques, the system can predict failure and issue the maintenance instructions, and then the jobs involved would respond to choose their own process resources and orders. Thus, the self-decision-making process could be realized.

Automated driving technology based on combining deep learning with incremental learning makes the logistics transport resourcing smart, so that vehicles and robots, can perceive their environment at real-time, choose the shortest route, and avoid congestion dynamically. When the information could be transferred between smart jobs and smart multi-resources, this decision-making process could be realized.

For constructing prediction models, some samples should be available. However before a real machining manufacturing is at work, there is no data. Therefore we have to turn to simulation technology. After building simulation models by the virtual simulation technology, a variety of working conditions are designed and simulated to obtain the required sample data. Based on the data obtained from the simulated and actual machining processes, the prediction model can be constantly improved by the deep learning and data-driven techniques.

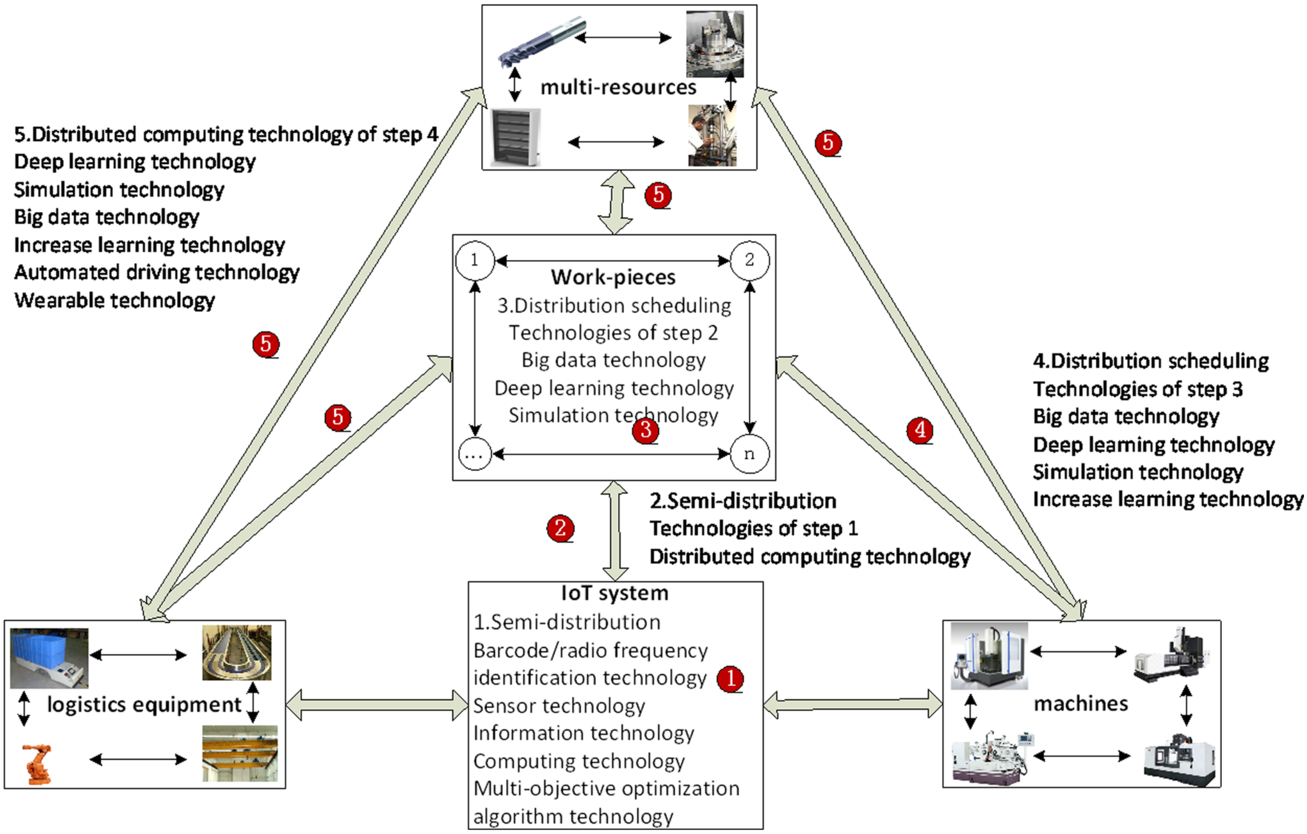

Fig. 4 Framework on solving JSP under Industry 4.0 with key enabling technologies

The implementation steps of JSP structure with smart factory (SFFJSP)

Considering a smart distributed scheduling within a smart factory consisting of multi-plants, work-pieces, machine tools and other resources, such as cutting tools, fixtures, logistics equipment and people, the original centralized scheduling problem can be turned into smart agents-based problems, which can be solved step by step with some key enabling technologies. The framework on solving the JSP problem under Industry 4.0 is shown on Fig. 4, which also describe the scheduling relationship among them.

According to the framework, we can implement the SFFJSP through five steps (shown in Table 3).

Step 1: Construction of Internet of things and network management system

With the barcode, radio frequency identification technology, sensor technology and so on, the connection is realized at real-time between the real factory and the synchronous dynamic simulation. Correctness and security of the planning and scheduling decision are verified by comparing the actual job shop statement and digital job shop statement by virtual simulation techniques. This step is the easiest work to be realized, but it prepares the necessary work and builds Internet of things and the gateway which is the foundation for a smart distributed scheduling platform. In this pre-stage, all information and data are still collected and controlled by the system platform. So after this step it is still only suitable for resolving a highly intelligent centralized job shop scheduling problem, but we can get all the information required for smart factory and then research how to pass the different information to different agents so that the different intelligent agents can complete their self-scheduling according to that information.

Step 2: Information is transferred between smart jobs and the system

Each job can obtain information from the system by passing the gateway of internet of things. When the jobs enter the system, they bring the machining information of production planning, NC code, process, processing time and the required resources, such as machine tools, vehicles, fixtures, tools, personnel etc., and then the system can obtain the information of the jobs. At the mean time the jobs can also obtain the information of the former entrancing jobs' scheduling and

Table 3 Five steps of SFFJSP

Step	Structure	Characteristics	Algorithm	Target	Key technology
1	Pre-stage: construction of Internet of things and network management system	Semi-distribution	Multi-objective Optimization algorithm	Operation sequence with selected machine, resource and plant	Barcode, radio frequency identification technology
		NP problem			Sensor technology
		Higher dynamic			Information technology
		Real-time			Computing technology
					Multi-objective optimization algorithm technology etc.
2	Information is transferred between smart jobs and the system	Semi-distribution	Multi-objective Optimization algorithm and distributed optimization algorithm: each job optimizes separately	Each job selects its own operation sequence with selected machine according to system judgment	Barcode, radio frequency identification technology
		Highest dynamic			Sensor technology
		Real-time			Information technology
		Self-organization			Multi-objective optimization algorithm technology
					Distributed computing technology
3	Information is transferred between smart jobs	Distribution	Distributed optimization algorithm: each job optimizes separately	Each job selects its own operation sequence with selected machine according to system rules	Barcode, radio frequency identification technology
		Highest dynamic			Sensor technology
		Real-time			Information technology
		Self-organization			Multi-objective optimization algorithm technology
		Self-adaptive			Distributed computing technology
		Self-learning			Big data technology
					Deep learning technology
					Simulation technology

Table 3 continued

Step	Structure	Characteristics	Algorithm	Target	Key technology
4	Information is transferred between smart jobs and smart machines	Distribution	Distributed optimization algorithm: jobs optimize separately	Each job selects its own operation sequence with selected machine	Barcode, radio frequency identification technology
		Highest dynamic	Each machine optimizes and selects job by itself		Sensor technology
		Real-time			Information technology
		Self-organization			Multi-objective optimization algorithm technology
		Self-adaptive			Distributed computing technology
		Self-diagnosis			Big data technology
		Self-learning			Deep learning technology
					Simulation technology
					Increase learning technology
5	Information is transferred between smart jobs and smart multi-resources	Distribution	Distributed optimization algorithm: jobs optimize separately	Each job selects its own operation sequence with selected machine, and other resources	Barcode, radio frequency identification technology
		Highest dynamic	Each resource optimizes and selects the job by itself		Sensor technology
		Real-time			Information technology
		Self-organization			Multi-objective optimization algorithm technology
		Self-diagnosis			Distributed computing technology
		Self-adaptive			Big data technology
		Self-learning			Deep learning technology
		Highest flexible			Simulation technology
					Increase learning technology
					Automated driving technology
					Wearable technology

logistics. According to the state and the rules, a smart job scheduling agent can choose intelligently the idle machine or the machine with the earliest completion time and realize self-scheduling. The information of the scheduling will be fed into the main control system, so that the system decides to agree or disagree according to the situation by considering the availability of a variety of processing resources, planning path uniformly, calculating the completion time and comparing with the production plan. If all meet the constrained conditions according to the analysis of the system, the scheduling is operated, otherwise rescheduling should be done until all of constraints are met. For the same jobs within the same batch, the job scheduling needs to be considered and calculated by the system or by some uniform rules, usually the first entrancing job.

Step 3: Information is transferred between smart jobs

If a job can avoid the busy area in accordance with the total information, the job can read other jobs' information of the "visible" scope in the idle zone directly to obtain their processing scheduling information. Then only considering the local constraint conditions, the job agent optimizes its own scheduling and then transfers the output information of optimization to the system and the jobs on the "visible" scope. The system calculates the busy and idle area for the next job, the other "visible" and unfinished scheduling jobs will complete their scheduling according to this information. This step is very important to the whole smart distributed scheduling. Without any centralized computing process, only need is to consider the rules of the system with some redundancy. Within a local range, a job as the main body of optimization, selects and assigns sequences and resources. Owing to the small amount of calculation, the exact solution can be obtained easier than the centralized scheduling. At the same time the smart scheduling based on the smart job (scheduling) agent can be achieved. In the initial stage, decision making is not necessarily an optimal solution for the entire system. But as time goes on, the system will feed back the relevant data after each decision. When data is accumulated to a certain extent, with the use of big data, depth learning and reinforcement learning, the self-decisions of the smart agents will become perfect. Compared to the centralized calculation and management, the way that each job chooses its machine tool and other resources is obviously simpler.

Step 4: Information is transferred between smart jobs and smart machines

Information transferred between jobs and machines can realize the real-time dynamic scheduling. When job information is passed to the available machines for operation, and if a machine tools judges that a job can be machined on it, it transfers the machine tools real-time information, including the state parameters of the machine tools and the earliest available time, the job then schedules and determines the earliest completion time. When a job needs to jump the queue, it is transported by the system to an available machine area, the emergency information is transmitted to the machine tools within a visual range, and then the available machines or the machines with the shortest finished time can be found. When the machine is broken down or with a predicted failure or in maintenance state, the information is transferred to the job agent in real time, and the job agent decides by itself to replace another machine. The centralized scheduling turns into the parallel intelligent distributed scheduling for both the jobs and machine tools.

Step 5: Information is transferred between smart jobs and smart multi-resources

Information transmission between the jobs and multi-resources can achieve a complete parallel intelligent and smart distributed scheduling. The information of cutting tools, fixtures, personnel and other information resources are all transferred to the machines. At the same time, the machines get the resources information required in the machining process, calculates the shortest preparation time of the resources, and issues a directive for requesting the scheduling of various resources so that a job can select the earliest available machine with a set of requested resources. Even the staff mood and health status can be considered in order to adjust the work intensity. With the automated driving technology, the smart agents of vehicles, robots and other transportation equipment can read the information from job agents and select the shortest path or the most unimpeded traffic flow path according to available machines and other resources. Simultaneously, with automatic collision avoidance in the "visible" range, the smart job can be transported to the smart machine with the shortest time. Therefore, a centralized scheduling can be changed into parallel distributed scheduling from three aspects: the smart machine, the smart job and the smart resources. Due to the distributed intelligent scheduling, the computation is greatly reduced, thus a more accurate optimal solution can be obtained and the optimal scheduling of the system can even be achieved by current algorithms at the beginning. However, after a period of the system operation, adding a variety of other techniques for big data analysis and intelligent decision-making, i.e. distributed computing technology, deep learning technology, simulation technology, incremental learning technology, automated driving technology, each smart agent can make decision directly and obtain the optimal scheduling with ease.

Conclusion

(1) JSP scheduling problems are summarized and reviewed, which are one of the most concerned problems currently in manufacturing.
(2) Different types of mathematical models according to their complexity and development trends are classified. And various algorithms used to solve the JSP models are also discussed along their development time line.
(3) New features of future JSP for Industry 4.0 are highlighted as smart distributed (or decentralized) scheduling, in contrast to the traditional centralized scheduling. Therefore the future trend of the JSP scheduling problem should lie in the development and utilization of the smart agent in the Industry 4.0. When the traditional centralized job shop scheduling is turned into the smart distributed scheduling, the computational workload can be greatly reduced and the system will become more flexible and agile.
(4) The actual job shop scheduling system is becoming more and more complex, dynamic and flexible. According to the development direction of the system, we not only need to consider JSP, but multi-machines, multi-resources, even a number of factories and logistics system. So there are four types of job shop scheduling, which are JSP, FJSP, MrFJSP and MpFJSP. However overwhelming majority of researchers still focus on the JSP and FJSP model although it is different from the actual complex production dramatically.
(5) After a job shop scheduling system is simplified and abstracted in its mathematical model, we can use all sorts of algorithms to solve the model. In spite of simplification, with the increasing of the complexity of the system, it is difficult to find a high quality solution in a short time. For the most of the simple JSP and FJSP model, scholars have to abandon the exact algorithm and choose the approximation algorithm to get a global solution or a sub optimal solution or only a non-dominated solution.
(6) Every year a large number of scholars devote themselves to the approximation algorithm for the JSP and its extensive study. In recent years, meta-heuristic algorithm or the combination of different algorithms are sought after. For example, the combination of meta-heuristic algorithm and local search method may gain a related ideal solution to the JSP problem. But the approximate algorithms often obtain the sub optimal solution, even the non-dominated solution owing to system complexity and simplification. Obviously, the traditional centralized or semi-centralized model not only limits the response time of the system, but also makes the calculation workload increased and it is difficult to converge to the optimal scheduling.
(7) Intelligent and smart distributed scheduling under Industry 4.0 is a key technology to solve this problem. Relying on the barcode, radio frequency identification technology and sensor technology, it achieves the real-time connection between an actual plant and its digital plant. By realizing direct or indirect information transmission and sharing among smart agents, it makes the different smart agents respectively to determine the different aspects of resources allocation and the static and dynamic scheduling. The centralized scheduling changes into a parallel distributed intelligent scheduling among the jobs, machines and resources, which plays a pivotal position to maximize the advantages of existing algorithms and the possibility of obtaining the exact solution.
(8) With the development of large data, high performance computing, cloud computing, deep learning and simulation technologies, different smart agents can predict results by themselves when scheduling is changed. With the prediction outputs, agents can construct their own multi-objective optimization models and solve them to achieve the best scheduling with current algorithms.

Acknowledgements This research is supported by the Aircraft Digital Workshop of Large-Scale Complex Structure Parts, Intelligent Manufacturing Special Project of Ministry of Industry and Information Technology of China (2015).

References

Aarts, E. H. L., & Lenstra, J. K. (Eds.). (1997). *Local search in combinatorial optimization*. London: Wiley.
Adams, J., Balas, E., & Zawack, D. (1988). The shifting bottleneck procedure for job shop scheduling. *Management Science*, *34*(3), 391–401.
Adibi, M. A., Zandieh, M., & Amiri, M. (2010). Multi-objective scheduling of dynamic job shop using variable neighborhood search. *Expert Systems with Applications*, *37*(1), 282–287.
Akyol, D. E., & Bayhan, G. M. (2007). A review on evolution of production scheduling with neural networks. *Computers & Industrial Engineering*, *53*(1), 95–122.
Apt, K. (2003). *Principles of constraint programming*. Cambridge: Cambridge University Press.
Balas, E., & Vazacopoulos, A. (1998). Guided local search with shifting bottleneck for job shop scheduling. *Management Science*, *44*(2), 262–275.
Barker, J. R., & McMahon, G. B. (1985). Scheduling the general job-shop. *Management Science*, *31*(5), 594–598.
Barták, R., Salido, M. A., & Rossi, F. (2010). Constraint satisfaction techniques in planning and scheduling. *Journal of Intelligent Manufacturing*, *21*(1), 5–15.
Baykasoğlu, A., Hamzadayi, A., & Köse, S. Y. (2014). Testing the performance of teaching-learning based optimization (TLBO) algorithm on combinatorial problems: Flow shop and job shop scheduling cases. *Information Sciences*, *276*, 204–218.
Blum, C. (2005). Beam-ACO—Hybridizing ant colony optimization with beam search: An application to open shop scheduling. *Computers & Operations Research*, *32*(6), 1565–1591.

Blum, C., & Sampels, M. (2004). An ant colony optimization algorithm for shop scheduling problems. *Journal of Mathematical Modelling and Algorithms*, *3*(3), 285–308.

Brooks, G. H., & White, C. R. (1965). An algorithm for finding optimal or near optimal solutions to production scheduling problem. *Journal of Industrial Engineering*, *16*(1), 34–40. https://www.researchgate.net/publication/284046228.

Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999). An improved ant system algorithm for the vehicle routing problem. *Annals of Operations Research*, *89*, 319–328.

Çaliş, B., & Bulkan, S. (2015). A research survey: Review of AI solution strategies of job shop scheduling problem. *Journal of Intelligent Manufacturing*, *26*(5), 961–973.

Canbolat, Y. B., & Gundogar, E. (2004). Fuzzy priority rule for job shop scheduling. *Journal of Intelligent Manufacturing*, *15*(4), 527–533.

Chen, Y. Y., Fu, L. C., & Chen, Y. C. (1998). Multi-agent based dynamic scheduling for a flexible assembly system. In *Proceedings, 1998 IEEE International Conference on Robotics and Automation* (Vol. 3, pp. 2122–2127). IEEE. doi:10.1109/ROBOT.1998.680634.

Chen, J. C., Wu, C. C., Chen, C. W., et al. (2012). Flexible job shop scheduling with parallel machines using genetic algorithm and grouping genetic algorithm. *Expert Systems with Applications*, *39*(11), 10016–10021.

Cheng, R., Gen, M., & Tsujimura, Y. (1999). A tutorial survey of job-shop scheduling problems using genetic algorithms. Part II: Hybrid genetic search strategies. *Computers & Industrial Engineering*, *36*(2), 343–364.

Chong, C. S., Sivakumar, A. I., Low, M. Y. H., et al. (2006). A bee colony optimization algorithm to job shop scheduling. In *Proceedings of the 38th conference on winter simulation. Winter simulation conference* (pp. 1954–1961). http://dl.acm.org/citation.cfm?id=1218469.

Colorni, A., Dorigo, M., & Maniezzo, V. (1991). Distributed optimization by ant colonies. In *Proceedings of the first European conference on artificial life* (Vol. 142, pp. 134–142). https://www.researchgate.net/publication/216300484.

Dauzere-Peres, S., & Lasserre, J. B. (1993). A modified shifting bottleneck procedure for job-shop scheduling. *The International Journal of Production Research*, *31*(4), 923–932.

Davis, L. (1985). Job shop scheduling with genetic algorithms. In *Proceedings of an international conference on genetic algorithms and their applications* (p. 140). Carnegie-Mellon University, Pittsburgh, PA.

Dorigo, M. (1992). *Optimization, learning and natural algorithms*. Ph.D. thesis, Politecnico di Milano, Italy.

Falkenauer, E., & Bouffouixm, S. (1991). A genetic algorithm for job shop. In *IEEE international conference on robotics and automation, Proceedings, 1991* (pp. 824–829). IEEE. doi:10.1109/ROBOT.1991.131689.

Fausett, L. (1994). *Fundamentals of neural networks: Architectures, algorithms, and applications*. Englewood Cliffs: Prentice-Hall.

Floudas, C. A., & Lin, X. (2004). Continuous-time versus discrete-time approaches for scheduling of chemical processes: A review. *Computers & Chemical Engineering*, *28*(11), 2109–2129.

Floudas, C. A., & Lin, X. (2005). Mixed integer linear programming in process scheduling: Modeling, algorithms, and applications. *Annals of Operations Research*, *139*(1), 131–162.

Fnaiech, N., Hammami, H., Yahyaoui, A., et al. (2012). New Hopfield neural network for joint job shop scheduling of production and maintenance. In *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society* (pp. 5535–5541). IEEE. doi:10.1109/IECON.2012.6389511.

Fonseca, D. J., & Navaresse, D. (2002). Artificial neural networks for job shop simulation. *Advanced Engineering Informatics*, *16*(4), 241–246.

French, S. (1982). *Sequencing and scheduling: An introduction to the mathematics of the job-shop*. New York: Ellis Horwood.

Garey, M. R., Johnson, D. S., & Sethi, R. (1976). The complexity of flowshop and job shops cheduling. *Mathematics of Operations Research*, *1*(2), 117–129.

Gen, M., & Lin, L. (2014). Multiobjective evolutionary algorithm for manufacturing scheduling problems: State-of-the-art survey. *Journal of Intelligent Manufacturing*, *25*(5), 849–866.

Geyik, F., & Cedimoglu, I. H. (2004). The strategies and parameters of tabu search for job-shop scheduling. *Journal of Intelligent Manufacturing*, *15*(4), 439–448.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, *13*(5), 533–549.

Glover, F., & Laguna, M. (2013). *Tabu search*. New York: Springer.

Graves, S. C. (1981). A review of production scheduling. *Operations Research*, *29*(4), 646–675.

Güçdemir, H., & Selim, H. (2017). Customer centric production planning and control in job shops: A simulation optimization approach. *Journal of Manufacturing Systems*, *43*, 100–116.

Harmanani, H. M., & Ghosn, S. B. (2016). An efficient method for the open-shop scheduling problem using simulated annealing (chapter). In *Information technology: New generations* (pp. 1183–1193). Berlin: Springer. doi:10.1007/978-3-319-32467-8_102.

Harrison, R. (2016) Engineering the smart factory. *Chinese Journal of Mechanical Engineering*, *1*. http://www.cjmenet.com/CN/article/downloadArticleFile.do?attachType=PDF&id=2905.

Hefetz, N., & Adiri, I. (1982). An efficient optimal algorithm for the two-machines unit-time jobshop schedule-length problem. *Mathematics of Operations Research*, *7*(3), 354–360.

Hermann, M., Pentek, T., & Otto, B. (2016). Design principles for Industrie 4.0 scenarios. In *2016 49th Hawaii international conference on system sciences (HICSS)* (pp. 3928–3937). IEEE. doi:10.1109/HICSS.2016.488.

Hino, R., & Moriwaki, T. (2002). Decentralized job shop scheduling by recursive propagation method. *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing*, *45*(2), 551–557.

Huang, S., Tian, N., & Ji, Z. (2016). Particle swarm optimization with variable neighborhood search for multi objective flexible job shop scheduling problem. *International Journal of Modeling, Simulation, and Scientific Computing.*. doi:10.1142/S1793962316500240.

Huang, K. L., & Liao, C. J. (2008). Ant colony optimization combined with taboo search for the job shop scheduling problem. *Computers & Operations Research*, *35*(4), 1030–1046.

Huang, R. H., Yang, C. L., & Cheng, W. C. (2013). Flexible job shop scheduling with due window—A two-pheromone ant colony approach. *International Journal of Production Economics*, *141*(2), 685–697.

Huang, R. H., & Yu, T. H. (2017). An effective ant colony optimization algorithm for multi-objective job-shop scheduling with equal-size lot-splitting. *Applied Soft Computing*, *57*, 642–656.

Ingimundardottir, H., & Runarsson, T. P. (2011). Supervised learning linear priority dispatch rules for job-shop scheduling. In *International conference on learning and intelligent optimization* (pp. 263–277). Berlin: Springer. http://link.springer.com/chapter/10.1007%2F978-3-642-25566-3_20.

Iwamura, K., & Sugimura, N. A. (2010). Study on real-time scheduling for autonomous distributed manufacturing systems. In *2010 IEEE international conference on systems man and cybernetics (SMC)* (pp. 1352–1357). IEEE. doi:10.1109/ICSMC.2010.5642451.

Jain, A. S., & Meeran, S. (1998). A state-of-the-art review of job-shop scheduling techniques. http://ftp.bstu.by/ai/To-dom/My_research/Paper-0-again/For-courses/Job-SSP/jain.pdf. Technical report, Department of Applied Physics, Electronic and Mechanical Engineering, University of Dundee, Dundee, Scotland.

Jalilvand-Nejad, A., & Fattahi, P. (2015). A mathematical model and genetic algorithm to cyclic flexible job shop scheduling problem. *Journal of Intelligent Manufacturing*, *26*(6), 1085–1098.

Johnson, S. M. (1954). Optimal two and three stage production schedules with setup times included. *Naval Research Logistics Quarterly*, *1*(1), 61–68. doi:10.1002/nav.3800010110/epdf.

Jones, D. F., Mirrazavi, S. K., & Tamiz, M. (2002). Multi-objective meta-heuristics: An overview of the current state-of-the-art. *European Journal of Operational Research*, *137*(1), 1–9.

Ju, Q. Y. (2007). *Planning and scheduling optimization of job-shop in intelligent manufacturing system* (pp. 1–97). Doctoral dissertation of Nanjing University of Aeronautics and Astronautics, Nanjing, China.

Kagermann, H. (2015). *Change through digitization—Value creation in the age of Industry 4.0. Management of permanent change* (pp. 23–45). Springer Fachmedien Wiesbaden, Wiesbaden. doi:10.1007/978-3-658-05014-6_2.

Kartam, N., & Tongthong, T. (1998). An artificial neural network for resource leveling problems. *Ai Edam*, *12*(3), 273–287.

Karthikeyan, S., Asokan, P., Nickolas, S., et al. (2015). A hybrid discrete firefly algorithm for solving multi-objective flexible job shop scheduling problems. *International Journal of Bio-Inspired Computation*, *7*(6), 386–401.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of IEEE international conference on neural network* (pp. 1942–1948), Perth. doi:10.1007/978-0-387-30164-8_630.

Khadwilard, A., Chansombat, S., Thepphakorn, T., Chainate, W., & Pongcharoen, P. (2012). Application of firefly algorithm and its parameter setting for job shop scheduling. *The Journal of Industrial Technology*, *8*(1), 49–58. https://www.researchgate.net/publication/225274007.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671–680. http://www.jstor.org/stable/1690046.

Kuczapski, A. M., Micea, M. V., Maniu, L. A., et al. (2015). Efficient generation of near optimal initial populations to enhance genetic algorithms for job-shop scheduling. *Information Technology and Control*, *39*(1). http://kalbos.ktu.lt/index.php/ITC/article/view/12091/6739.

Li, X., & Gao, L. (2016). An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem. *International Journal of Production Economics*, *174*, 93–110.

Lian, L., & Mesghouni, K. (2014). Comparative study of heuristics algorithms in solving flexible job shop scheduling problem with condition based maintenance. *Journal of Industrial Engineering and Management*, *7*(2), 518–531.

Lin, L., Hao, X. C., Gen, M., et al. (2012). Network modeling and evolutionary optimization for scheduling in manufacturing. *Journal of Intelligent Manufacturing*, *23*(6), 2237–2253.

Liu, S. Q., & Kozan, E. (2012). A hybrid shifting bottleneck procedure algorithm for the parallel-machine job-shop scheduling problem. *Journal of the Operational Research Society*, *63*(2), 168–182.

Lomnicki, Z. A. (1965). A "branch-and-bound" algorithm for the exact solution of the three-machine scheduling problem. *Journal of the Operational Research Society*, *16*(1), 89–100.

Łukasik, S., & Żak, S. (2009). Firefly algorithm for continuous constrained optimization tasks. In *International conference on computational collective intelligence* (pp. 97–106). Berlin: Springer. doi:10.1007/978-3-642-04441-0_8.

Manne, A. S. (1960). On the job-shop scheduling problem. *Operations Research*, *8*(2), 219–223.

Marichelvam, M. K., & Geetha, M. (2016). A hybrid discrete firefly algorithm to solve flow shop scheduling problems to minimise total flow time. *International Journal of Bio-Inspired Computation*, *8*(5), 318–325.

Marichelvam, M. K., Prabaharan, T., & Yang, X. S. (2014). A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems. *IEEE Transactions on Evolutionary Computation*, *18*(2), 301–305.

McMahon, G., & Florian, M. (1975). On scheduling with ready times and due dates to minimize maximum lateness. *Operations Research*, *23*(3), 475–482.

Meeran, S., & Morshed, M. S. (2012). A hybrid genetic tabu search algorithm for solving job shop scheduling problems: A case study. *Journal of Intelligent Manufacturing*, *23*(4), 1063–1078.

Merkle, D., Middendorf, M., & Schmeck, H. (2002). Ant colony optimization for resource-constrained project scheduling. *IEEE Transactions on Evolutionary Computation*, *6*(4), 333–346.

Min, H. S., & Yih, Y. (2003). Selection of dispatching rules on multiple dispatching decision points in real-time scheduling of a semiconductor wafer fabrication system. *International Journal of Production Research*, *41*(16), 3921–3941.

Morton, T., & Pentico, D. W. (1993). *Heuristic scheduling systems: With applications to production systems and project management*. London: Wiley.

Muthiah, A., Rajkumar, A., & Rajkumar, R. (2016). Hybridization of artificial bee colony algorithm with particle swarm optimization algorithm for flexible job shop scheduling. In *2016 international conference on energy efficient technologies for sustainability (ICEETS)* (pp. 896–903). IEEE.

Nawaz, M., Enscore, E. E., & Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, *11*(1), 91–95.

Neto, R. F. T., & GodinhoFilho, M. (2013). Literature review regarding ant colony optimization applied to scheduling problems: Guidelines for implementation and directions for future research. *Engineering Applications of Artificial Intelligence*, *26*(1), 150–161.

Nouiri, M., Bekrar, A., Jemai, A., et al. (2015). An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-015-1039-3.

Nouiri, M., Bekrar, A., Jemai, A., et al. (2017). Two stage particle swarm optimization to solve the flexible job shop predictive scheduling problem considering possible machine breakdowns. *Computers & Industrial Engineering*. doi:10.1016/j.cie.2017.03.006.

Paul, M., Sridharan, R., & Ramanan, T. R. (2016). A multi-objective decision-making framework using preference selection index for assembly job shop scheduling problem. *International Journal of Management Concepts and Philosophy*, *9*(4), 362–387.

Peng, B., Lü, Z., & Cheng, T. C. E. (2015). A tabu search/path relinking algorithm to solve the job shop scheduling problem. *Computers & Operations Research*, *53*, 154–164.

Pesch, E., & Tetzlaff, U. A. W. (1996). Constraint propagation based scheduling of job shops. *INFORMS Journal on Computing*, *8*(2), 144–157.

Pham, D. T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., & Zaidi, M. (2005). *The bees algorithm*. Technical note, Manufacturing Engineering Center, Cardiff University.

Ponsich, A., & Coello, C. A. C. (2013). A hybrid differential evolution—Tabu search algorithm for the solution of job-shop scheduling problems. *Applied Soft Computing*, *13*(1), 462–474.

Ponsich, A., Tapia, M. G. C., & Coello, C. A. C. (2009). Solving permutation problems with differential evolution: An application to the jobshop scheduling problem. In *ISDA* (pp. 25–30). doi:10.1109/ISDA.2009.49.

Potts, C. N., & Van Wassenhove, L. N. (1985). A branch and bound algorithm for the total weighted tardiness problem. *Operations Research*, *33*(2), 363–377.

Reeves, C. R. (1993). *Modern heuristic techniques for combinatorial problems*. London: Wiley.

Rosenkrantz, D. J., Stearns, R. E., & Lewis, P. M, I. I. (1977). An analysis of several heuristics for the traveling salesman problem. *SIAM Journal on Computing*, *6*(3), 563–581.

Rossi, F., Van Beek, P., & Walsh, T. (2006). *Handbook of constraint programming*. Amsterdam: Elsevier. http://www.informatik.uni-ulm.de/pm/fileadmin/pm/home/fruehwirth/Papers/FAI.pdf.

Sadeh, N., & Fox, M. S. (1996). Variable and value ordering heuristics for the job shop scheduling constraint satisfaction problem. *Artificial Intelligence*, *86*(1), 1–41.

Saidi-Mehrabad, M., Dehnavi-Arani, S., Evazabadian, F., et al. (2015). An ant colony algorithm (ACA) for solving the new integrated model of job shop scheduling and conflict-free routing of AGVs. *Computers & Industrial Engineering*, *86*, 2–13.

Sakawa, M., & Kubota, R. (2000). Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy duedate through genetic algorithms. *European Journal of Operational Research*, *120*(2), 393–407.

Sarin, S. C., Ahn, S., & Bishop, A. B. (1988). An improved branching scheme for the branch and bound procedure of scheduling n jobs on m parallel machines to minimize total weighted flowtime. *The International Journal of Production Research*, *26*(7), 1183–1191.

Sayadi, M., Ramezanian, R., & Ghaffari-Nasab, N. (2010). A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems. *International Journal of Industrial Engineering Computations*, *1*(1), 1–10.

Shivasankaran, N., Kumar, P. S., & Raja, K. V. (2015). Hybrid sorting immune simulated annealing algorithm for flexible job shop scheduling. *International Journal of Computational Intelligence Systems*, *8*(3), 455–466.

Simon, F. Y. P., & Takefuji, Y. (1988). Stochastic neural networks for solving job-shop scheduling. I. Problem representation. In *IEEE international conference on neural networks* (pp. 275–282). IEEE. doi:10.1109/ICNN.1988.23939.

Singh, M. R., & Mahapatra, S. S. (2016). A quantum behaved particle swarm optimization for flexible job shop scheduling. *Computers & Industrial Engineering*, *93*, 36–44.

Sotskov, Y. N., Tautenhahn, T., & Werner, F. (1999). On the application of insertion techniques for job shop problems with setup times. *RAIRO-Operations Research*, *33*(2), 209–245.

Storn, R., & Price, K. (1995). *Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces*. Berkeley: ICSI.

Teekeng, W., Thammano, A., Unkaw, P., et al. (2016). A new algorithm for flexible job-shop scheduling problem based on particle swarm optimization. *Artificial Life and Robotics*, *21*(1), 18–23.

Van Dyke Parunak, H., Irish, B. W., Kindrick, J., & Lozo, P. W. (1985). Fractal actors for distributed manufacturing control. In Second conference on artificial intelligence applications: The engineering of knowledge-based systems (pp. 653–660). Miami Beach, FL, USA.

Van Eck, M. (2003). *Advanced planning and scheduling—Is logistics everything?* Amsterdam: University of Amsterdam. http://www.docin.com/p-1372360378.html.

Wagner, H. M. (1959). An integer linearprogramming model for machine scheduling. *Naval Research Logistics Quarterly*, *6*(2), 131–140.

Wang, L., Cai, J., Li, M., et al. (2017). Flexible job shop scheduling problem using an improved ant colony optimization. *Scientific Programming*. doi:10.1155/2017/9016303.

Wang, C., & Jiang, P. (2016). Manifold learning based rescheduling decision mechanism for recessive disturbances in RFID-driven job shops. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-016-1194-1.

Wenqi, H., & Aihua, Y. (2004). An improved shifting bottleneck procedure for the job shop scheduling problem. *Computers & Operations Research*, *31*(12), 2093–2110.

Werner, F., & Winkler, A. (1995). Insertion techniques for the heuristic solution of the job shop problem. *Discrete Applied Mathematics*, *58*(2), 191–211.

Xanthopoulos, A. S., & Koulouriotis, D. E. (2015). Cluster analysis and neural network-based metamodeling of priority rules for dynamic sequencing. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-015-1090-0.

Xia, W., & Wu, Z. (2005). An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. *Computers & Industrial Engineering*, *48*(2), 409–425.

Xing, L. N., Chen, Y. W., Wang, P., et al. (2010). A knowledge-based ant colony optimization for flexible job shop scheduling problems. *Applied Soft Computing*, *10*(3), 888–896.

Xue, H., Zhang, P., & Wei, S. (2015). Applying a hybrid algorithm of immunity and ant colony in job-shop scheduling. In *Industrial engineering and manufacturing technology: Proceedings of the 2014 international conference on industrial engineering and manufacturing technology (ICIEMT 2014)* (Vol. 4, p. 91), Shanghai, China. CRC Press, 10–11 July 2014.

Yang, X. S. (2008). *Nature-inspired metaheuristic algorithms*. Bristol: Luniver Press.

Yin, L., Yang, L., & Hu, M. (2015). Job shop scheduling based on improved discrete particle swarm optimization. In *Proceedings of the 21st international conference on industrial engineering and engineering management 2014* (pp. 99–101). Amsterdam: Atlantis Press. doi:10.2991/978-94-6239-102-4_21.

Zahmani, M. H., Atmani, B., Bekrar, A., et al. (2015). Multiple priority dispatching rules for the job shop scheduling problem. In *2015 3rd international conference on control, engineering and information technology (CEIT)* (pp. 1–6). IEEE. doi:10.1109/CEIT.2015.7232991.

Zandieh, M., Khatami, A. R., & Rahmati, S. H. A. (2017). Flexible job shop scheduling under condition-based maintenance: Improved version of imperialist competitive algorithm. *Applied Soft Computing*, *58*, 449–464.

Zhang, R., & Chong, R. (2016). Solving the energy-efficient job shop scheduling problem: A multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption. *Journal of Cleaner Production*, *112*, 3361–3375.

Zhang, W., Wen, J. B., Zhu, Y. C., et al. (2017). Multi-objective scheduling simulation of flexible job-shop based on multi-population genetic algorithm. *International Journal of Simulation Modelling (IJSIMM)*. doi:10.2507/IJSIMM16(2)CO6.

Zhang, H., Yan, Q., Zhang, G., et al. (2016). A chaotic differential evolution algorithm for flexible job shop scheduling. In *Asian simulation conference* (pp. 79–88). Singapore: Springer.

Zhao, B., Gao, J., Chen, K., et al. (2015). Two-generation Pareto ant colony algorithm for multi-objective job shop scheduling problem with alternative process plans and unrelated parallel machines. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-015-1091-z.

Zhao, F., Shao, Z., Wang, J., et al. (2016). A hybrid differential evolution and estimation of distribution algorithm based on neighbourhood search for job shop scheduling problems. *International Journal of Production Research*, *54*(4), 1039–1060.

Zorin, D. A., & Kostenko, V. A. (2014). Simulated annealing algorithm for job shop scheduling on reliable real-time systems. In *International conference on operations research and enterprise systems* (pp. 31–46). Berlin: Springer. doi:10.1007/978-3-319-17509-6_3.