Front. Eng. Manag. 2023, 10(3): 406–426
https://doi.org/10.1007/s42524-022-0242-0

REVIEW ARTICLE

Yuting WANG, Yuyan HAN, Dunwei GONG, Huan LI

A review of intelligent optimization for group scheduling problems in cellular manufacturing

Abstract Given that group technology can reduce the changeover time of equipment, broaden the productivity, and enhance the flexibility of manufacturing, especially cellular manufacturing, group scheduling problems (GSPs) have elicited considerable attention in the academic and industry practical literature. There are two issues to be solved in GSPs: One is how to allocate groups into the production cells in view of major setup times between groups and the other is how to schedule jobs in each group. Although a number of studies on GSPs have been published, few integrated reviews have been conducted so far on considered problems with different constraints and their optimization methods. To this end, this study hopes to shorten the gap by reviewing the development of research and analyzing these problems. All literature is classified according to the number of objective functions, number of machines, and optimization algorithms. The classical mathematical models of single-machine, permutation, and distributed flowshop GSPs based on adjacent and position-based modeling methods, respectively, are also formulated. Last but not least, outlooks are given for outspread problems and problem algorithms for future research in the fields of group scheduling.

Keywords cellular manufacturing, group scheduling, flowshop, literature review

Received April 2, 2022; accepted September 6, 2022

Yuting WANG, Yuyan HAN (✉), Huan LI
School of Computer Science, Liaocheng University, Liaocheng 252059, China
E-mail: hanyuyan@lcu-cs.com

Dunwei GONG
School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China

This work is partially supported by the National Natural Science Foundation of China (Grant Nos. 61803192, 61966012, 61973203, and 62106073), Guangyue Young Scholar Innovation Team of Liaocheng University (Grant No. LCUGYTD2022-03), the Youth Innovation Talent Introduction and Education Program support from Shandong Province Colleges and Universities, the Natural Science Foundation of Hunan Province (Grant No. 2021JJ40116), and the Natural Science Foundation of Shandong Province (Grant Nos. ZR2021QE195 and ZR2021QF036).

1 Introduction

In cellular manufacturing, i.e., printed circuit boards (PCBs), different PCB types can be classified into PCB families. In the preparing stage, chips inserted on all the PCBs in a family are preloaded on the insertion machines (Schaller et al., 2000). If different PCB families need a switchover, for the new family, the number of different chips loaded on the machines has two cases to be considered: Chips common between families were not changed to reduce the changeover time, and chips not common between families should be changed. Thus, the changeover time depends on the sequence. In this context, the scheduling problem was a sequence-dependent group/family scheduling problem for a pure flow line in cellular manufacturing (Mahmoodi and Dooley, 1991; Logendran, 1992; Ruben et al., 1993; Yang and Liao, 1996; Logendran et al., 2006b).

Based on this application scene, group technology is proposed to broaden the productivity and enhance the flexibility of manufacturing. Especially in the assembly of PCBs of cellular manufacturing scheduling, the sets of products yielded by each manufacturing cell are considered job families. This is, jobs that have the same resources or the similarity in designing or processing are grouped into a family, forming a new scheduling problem, called the group scheduling problem (GSP). Owing to group technology having the advantages of efficient, low-cost production, and simple structure, it is clearly emerging as one of the most important manufacturing modes in modern enterprises, including Huawei Machine Company Ltd., Foxconn Technology Group, and Zhongtong Bus Holding Company Ltd. (Logendran, 1992). The difference between the traditional flowshop scheduling problems

(FSP) and GSP is that the former only considers how to schedule jobs in the manufacturing cells; however, the latter has two key issues to be solved, i.e., how to allocate groups into the production cells in view of major setup times between groups and how to schedule jobs in each group. Thus, GSP is more complicated in property analysis and algorithm designing than the traditional FSPs and has important practical and theoretical significance.

Owing to the complexity of GSP, designing efficient scheduling algorithms to improve the competitiveness for responding to marketable changes is essential. Thus, the model construction of GSP and its scheduling algorithms have been simultaneously considered in the academic field and manufacturing. GSPs are split into four types according to the technological process or the technological constraints in manufacturing: Single-machine GSPs, parallel-machine GSPs, permutation flowshop GSPs (PFGSPs), and distributed flowshop GSPs (DFGSPs). In addition to these four types, some literature reviews the related GSPs or group technology in other practical applications, such as networks and reconfigurable manufacturing systems (RMSs). Zhang et al. (2019) designed a framework for the reconfigurable digital twin system, in which a multi-mode perception sensor group is considered in the physical layer. Li et al. (2018) introduced a real-time decentralized management framework where reconfigurable machines are grouped to form reconfigurable manufacturing lines. Tang et al. (2022b) proposed a deep reinforcement learning (RL) approach to solve RMS scheduling. They adopted a group of deep RL agents to find a dynamic control policy and embedded these agents with a shared value decomposition network. For the modeling of RMS architecture, Gu (2022) built a discrete-time Markov chain model to perform an exact analysis for two-stage-one-buffer systems.

For the literature survey, we consult online journal websites, i.e., Elsevier, Taylor & Francis, and Springer, using the corresponding keywords. A detailed search has been done on Web of Science and Scopus, using the following keywords: Group technology, group flowshop scheduling, flowshop group scheduling, job shop group scheduling, group scheduling, and group production scheduling. We also benefited from the recent survey of Neufeld et al. (2016), which considers all publications until 2015. We tried to cover all group manufacturing scheduling studies belonging to the last decade. By retrieving from Elsevier, Taylor & Francis, and Springer, we find 109 articles that are closely related to GSPs. Figure 1 gives the number of publications per year since 1990. It can be observed that the number of papers on GSPs is relatively small before 2010 (i.e., between 1990 and 2009). Clearly, the articles about GSPs have increased since 2010, suggesting that the problems have become a hot topic in the related fields. Figure 2 shows the number of papers published in each journal, and Fig. 3 displays a word cloud diagram showing the distribution of the papers in the journals. The majority of research in group scheduling is published in *International Journal of Production Research*, *Computers & Industrial Engineering*, and *International Journal of Production Economics*.

Although many studies on GSPs have been published, few integrated reviews on considered problems with different constraints and their optimization methods have been conducted so far. To this end, first, the literature is classified, according to the characteristic of the workshop configuration, into two aspects, i.e., single- and parallel-machine GSPs and flowshop GSPs (FGSPs), in which all literature is further classified according to the number of objective functions, the number of machines, and the optimization algorithms. Then, the classical mathematical models of single-machine GSPs, and permutation and distributed FGSPs based on adjacent and position-based modeling methods, respectively, are formulated. Finally, we summarize and analyze them, and state several future research trends. This study hopes to shorten the gap by

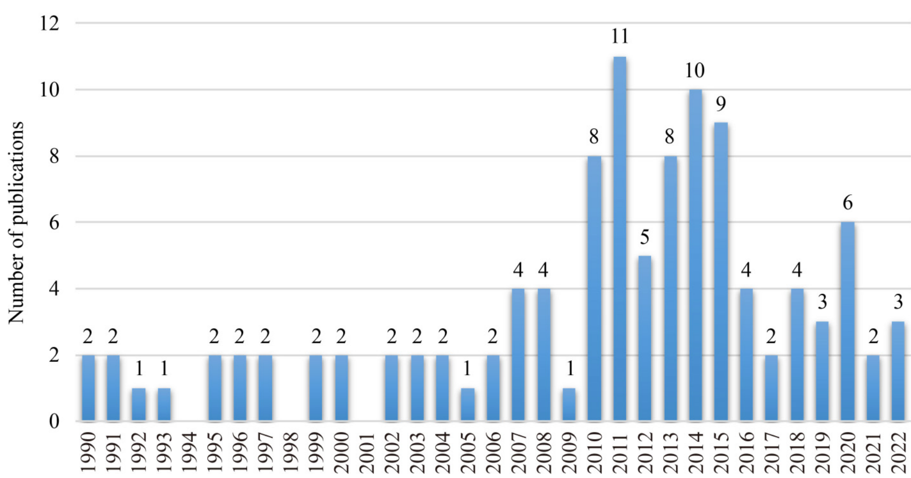

Fig. 1 Number of publications per year.

Fig. 2 Number of papers published in each journal.

reviewing the development of research and analyzing the above problems.

The remainder of this paper is organized as follows. Section 2 reviews some existing literature that solves the related problems. In Section 3, notations, mathematical models, and existing benchmark instances are stated. Section 4 analyses research survey results. Section 5 gives future research trends.

2 Literature review

The GSPs existing in cellular manufacturing have been described by many researchers. According to the constraints of technological process, the two main categories of GSPs have been investigated respectively. In addition, some intelligent optimization algorithms for solving discrete scheduling problems similar to group scheduling ones have been also elaborated, which can help the readers refer to the intelligent optimization algorithms to solve GSPs better.

2.1 Single- and parallel-machine GSPs

The following describes the definition of the classical single GSP (Janiak and Kovalyov, 1995): n jobs are processed on a single machine. Jobs are classified into g groups or families according to the group technology. For jobs from the same group, they must be processed uninterruptedly and no setup time exists between any two adjacent jobs. However, a setup time exists between the different groups (Biskup, 1999).

The single-machine group scheduling is a sub-question of GSPs and has been widely used in industrial production. Yazdani Sabouni and Logendran (2013a) analyzed the application of the PCB problem and developed a branch-and-bound algorithm based on the lower-bounding (LB) structure. Janiak and Kovalyov (1995) defined the quality of the solution from several aspects for the single-machine GSP and demonstrated that the considered problem is polynomially solvable. In this work, for the traditional single-machine GSP, the processing time and setup time are constant. In real production activities, the times

Fig. 3 Word cloud diagram of journal names.

of processing and setup always change because of the deteriorating jobs and learning effect. According to the case, the processing time of the linear decreasing and increasing functions are studied, and the problem is proved to be polynomially solvable in both cases (Wang et al., 2007; 2009). After that, Wang and Wang (2014) analyzed the relation between the processing time and setup time by a time linear function. Liu et al. (2010) studied the deteriorating jobs and grouping technology separately, proposed the two-agent scheduling problems, and optimized the total completion time and maximum cost, respectively.

Considering the learning effect in actual production, Lee and Wu (2009) proposed a novel learning model of group scheduling. On this basis, the same problem is expanded according to the actual production, and the problem is proved by a solvable polynomial in this situation (Yang and Yang, 2010; Yang, 2011). Bai et al. (2012) studied degradation and learning effects with minimal makespan. Kuo (2012) considered learning effects in different situations and designed two polynomial time algorithms (PTAs) to solve different objectives. Low and Lin (2012) considered a past-sequence-dependent (PSD) setup time and a learning effect in the GSP of the single machine. Pan et al. (2014) combined forgetting and learning effects and added preventive maintenance (PM) plans for their study, and constructed a new mathematical model. According to the characteristic of the problem, the authors also developed the search algorithm for the model. Zhang et al. (2018) designed some novel models based on position-dependent processing times and employed a polynomial-time algorithm.

Combined with the actual conditions of the factory, the delivery date of the factory is also a factor worth considering. Li et al. (2011) added an expiration date assignment target. Lu et al. (2014) set release date minimization as the goal of optimization. Keshavarz et al. (2015b) studied the single-machine scheduling problem with minimal total weighted earliness and tardiness. In this work, a branch-and-bound algorithm based on Lagrangian is employed to optimize the above objective. Li and Zhao (2015) considered the multiple due windows assignment constraint and minimized the two objectives, i.e., the due windows related costs and total of earliness and tardiness. Yue et al. (2016) optimized multiple objectives such as makespan and total weighted delay time and proposed an improved hybrid Pareto artificial bee colony (HPABC) algorithm. Nie et al. (2007) designed a prefix-gene expression programming algorithm to optimize the earliness and tardiness penalties. To balance the conflict relation between setup time decreasing and delivery satisfaction, a variable heuristic search and look-ahead constraint propagation method was employed (Jiang et al., 2013).

In addition to the makespan, tardiness time, and earliness time, the resources allocated have been researched. Yan and Zhao (2007) considered three resource consumptions of the single-machine GSPs with continuous resources and constructed three mathematical models. Zhu et al. (2011) considered the learning effect and resource allocation in single-machine GSP. In this work, the functions with learning effect and two resource allocation have verified that the problems remain polynomially solvable.

Huang et al. (2011) have added resource-constrained constraints considering learning effects and degradation and set the objective to minimize total resources and makespan. Two different questions with different objectives have also been optimized under resource constraints (Huang and Wang, 2014). For the single-machine GSP with deteriorating jobs, learning effect and resource allocation, the solvability of the problem in polynomial time is proved by Yin et al. (2014). Wang and Liu (2014) considered two objectives about total weighted value of maximum completion time and maximum cost and adopted a PTA. Yang et al. (2008b) discussed characterizations of the optimal schedules and presented the optimal allocation methods.

These papers are summarized in Table 1. For simplification, the two objectives of makespan and the total completion time are abbreviated to M&TCT, the objective of weighted sum of makespan and total resource cost is abbreviated to WMTRC, and the weighted sum of total completion time and total resource cost is WTCTTRC for short. SPT and LPT refer to the smallest and longest normal processing time first rule, respectively.

In conclusion, the single- and parallel-machine GSPs are solved by using the heuristic and accurate algorithms, i.e., PTA, SPT, LPT, and branch-and-bound algorithm based on Lagrangian, for the small instances. For the large instances, meta-heuristic approaches, i.e., search algorithm, genetic algorithm (GA), and HPABC, are used most commonly to find approximately optimal solutions. For the single- and parallel-machine GSPs, the number of machines in stages is small. The heuristics or constructive heuristics are considered to solve the above problems because they require significantly lower computation complexity and can obtain good solutions in a short time. However, for the multi-stage or m-machine FSPs, the single heuristic cannot effectively optimize these problems. Thus, developing intelligent optimization algorithms is still needed.

2.2 Flowshop GSPs

The traditional FGSP is an important FSP in manufacturing

Table 1 Single- and parallel-machine GSPs

Reference	Objective		Algorithm		Model
	Single objective	Multi-objective	Heuristic	Meta-heuristic	
Wang et al. (2007)		M&TCT		PTA	
Wang et al. (2009)	Makespan				√
Lee and Wu (2009)		M&TCT		PTA	√
Liu et al. (2010)		First agent: Total completion time Second agent: Maximum cost		PTA	
Yang and Yang (2010)	Total completion time		SPT	PTA	√
Li et al. (2011)	Cost function			$O(n \log n)$ time unified optimization algorithm	√
Zhu et al. (2011)		WMTRC and WTCTTRC		PTA	√
Huang et al. (2011)		WMTRC			√
Yang (2011)		M&TCT	SPT, LPT	PTA	√
Bai et al. (2012)		M&TCT			√
Low and Lin (2012)		M&TCT	SPT		√
Kuo (2012)		M&TCT		PTA	√
Yazdani Sabouni and Logendran (2013a)	Makespan			Branch-and-bound algorithm	
Huang and Wang (2014)		Makespan and resource consumption			√
Wang and Wang (2014)		Makespan			√
Pan et al. (2014)	Makespan			Search algorithm, GA	√
Wang and Liu (2014)		Total weighted completion time and maximum cost		PTA	
Yin et al. (2014)	WMTRC			PTA	√
Lu et al. (2014)	Makespan			PTA	√
Keshavarz et al. (2015b)	Sum of earliness and tardiness penalties		Local search algorithms	Branch-and-bound algorithm based on Lagrangian	√
Yue et al. (2016)		Makespan and the total weighted tardiness time		HPABC	√
Zhang et al. (2018)		M&TCT		PTA	√

fields. FGSP has been verified to be a non-deterministic polynomial (NP)-hard problem (Schaller et al., 2000) that has been extensively studied by many researchers (Yoshida et al., 1977; Schaller, 2001; Logendran et al., 2005). The definition of FGSP is given as follows: A collection of n jobs must pass through all predetermined m stages. At each stage, there is a machine to handle the upcoming job. All jobs should process on all the machines with the same path. Before processing, each job has its own assigned, affiliated family or group g. The sequence-dependent setup times between different families are considered at all machines. Notably, there is no setup time between jobs belonging to the same family, or this time is included in the job's processing time. In addition, the jobs from different families cannot be intermingled with each other. In other words, if the affiliated family of one job is assigned and determined, no matter what scheduling methods and algorithms are used, the jobs of this family cannot be separated and put into other families. Moreover, all jobs are available at time zero and before starting to be processed on the machine. Up to now, the FGSP has emerged in many real-world problems, such as airplane engine blades (Schaller, 2001), label sticker manufacturing (Li, 1997), furniture production (Lin and Liao, 2003), bridge construction (Wilson et al., 2004), electronics manufacturing (Yang et al., 2008a), automotive paint and body shops (Gelogullari and Logendran, 2010), metal parts punch (Salmasi et al., 2010), and thin film transistor (TFT)-liquid crystal display (LCD) (van der Zee, 2013). Therefore, the study of FGSP has not only academic significance but also practical application value.

As far as we know, at present, in the literature on solving FGSPs, the optimization objectives of the problems are generally classified into two types: Single- and multi-objective optimization. Between them, more studies have been made on optimizing single-objective FGSP, i.e., makespan and total flow time. Heuristics methods are used to generate good initialization solutions based on the rule of problems (Radharamanan, 1986; Allison, 1990; Mahmoodi et al., 1990; Logendran and Nudtasomboon, 1991; Logendran et al., 1995; Frazier, 1996; Yang, 2002; Salmasi and Logendran, 2008; Villadiego et al., 2012; Bozorgirad and Logendran, 2016). However, for the medium- and large-scale GSP, the performance of heuristics is low. Thus, heuristics methods are embedded into intelligent algorithms to enhance the quality of the initialization solution. For the makespan criterion, Salmasi et al. (2011) and Keshavarz and Salmasi (2013) constructed the mathematical model of the FGSP, then presented a hybrid ant colony optimization (HACO) algorithm. In these studies, a mathematical model based on an LB technique was adopted to estimate the property of the HACO. In another study (Yang and Chern, 2000), a transportation time is considered, and a PTA is employed to solve GSP with makespan. Keshavarz et al. (2015a) adopted a metaheuristic algorithm based on the memetic algorithm for GSP, and the results showed that the average percentage difference of the proposed algorithm is 6.03%. To decrease the costs of manufacturers, the role of the kitting staff must be eliminated. In another work (Yazdani Sabouni and Logendran, 2013b), the external setup time is considered for the next board group and required to be performed by the machine operator.

For minimizing the total flow time of job sequence, Hajinejad et al. (2011) suggested a particle swarm optimization (PSO) algorithm with an encoding scheme based on ranked order value to convert the particle position value to the job and group permutations. Costa et al. (2014) proposed a hybrid metaheuristic algorithm to optimize the total flowshop criterion of flowshop sequence-dependent GSP. Later, Mendes et al. (2013) developed a hybrid heuristic based on variable neighborhood descent (VND) and iterated local search (ILS) metaheuristic to solve the FGSP by minimizing the total flow time. For the same objective, Keshavarz and Salmasi (2014) developed the hybrid genetic (HG) algorithm and LB method according to the strong NP-hard of FGSP to find a better scheduling sequence. Based on the traditional characteristics of the FGSP, Khamseh et al. (2015) extended and integrated this problem to be a flexible FGSP with sequence-dependent setups. In this literature, two metaheuristics, i.e., simulated annealing (SA) and GA are designed to optimize the makespan value.

Zolfaghari and Liang (1999) designed a hybrid algorithm based on Tabu search (TS) and SA approach to solve the group scheduling and machining speed selection problems. Next, Cho and Ahn (2003) addressed an HG algorithm to solve the group scheduling with sequence-dependent setup time. Liou and Liu (2010), Liou et al. (2013) and Liou and Hsieh (2015) presented PSO and hybrid algorithms to solve GSP, respectively. For flexible flowshop sequence-dependent GSP, some intelligence algorithms, i.e., TS and imperialist competitive algorithm, are developed (Logendran et al., 2006a; Karimi et al., 2011; Shahvari et al., 2012). In recent years, as the research progresses, more complex problems and algorithms have been gradually proposed. Zandieh and Hashemi (2015) considered the stochastic breakdowns' assumptions that make a dynamic nature in the hybrid flexible FGSP and used the GA to optimize the expected average values of completion times. Neufeld et al. (2015) presented two levels of strategies, in which the job sequence is generated with minimal inserted machine idle times rather than makespan, and a significant solution can be obtained by using a scheduling method. Adressi et al. (2016) simultaneously considered the no-wait conditions and random breakdown of machines in flexible FGSP and proposed the GA and SA algorithms to optimize the makespan. Behjat and Salmasi (2017) presented a mixed integer linear mathematical model of the no-wait FGSP and developed some meta-heuristics based on the PSO algorithm and VND structures for solving this problem. Feng et al. (2019) designed a new mixed integer linear programming (MILP) model by combining PM and flexible FGSP. The work contains the relaxation of group technology assumptions, the setup times between different families, and the integration of PM and group. In addition, for the no-wait condition, Lin and Ying (2019) used an efficient meta-heuristic to solve the FGSP in a reasonable computational time, and the comparison results have demonstrated that the proposed algorithm is superior to those of state-of-the-art meta-heuristic algorithms.

For the FGSP with round-trip transportation time between machines, Yuan et al. (2021) proposed the MILP model and designed a co-evolutionary discrete differential evolution algorithm (CDDEA) to optimize the makespan of the scheduling sequence. Subsequently, other constraints are considered in the FGSP. Costa et al. (2020) considered the blocking constraints and developed an original meta-heuristic approach to solve the blocking FGSP by minimizing the makespan. Then, Yuan et al. (2020a) considered the blocking constraint and transportation times of jobs in FGSP and constructed an MILP model. To solve this model, the authors presented a co-evolutionary GA (CGA). From the experimental results, it can be seen that the constraints considered in GSP are becoming more and more complex than those in the traditional FSPs. Furthermore, the GSP has been extended to distributed environments. Pan et al. (2022) designed an MILP model of the distributed FGSP, a cooperative co-evolutionary algorithm (CCEA) with a reinitialization scheme and a collaboration model to optimize the makespan. The above research shows that many constraints are gradually considered in the problem to make it more in line with real-world conditions. For the same DFGSP, Wang et al. (2022) constructed the mathematical model and designed an effective two-stage iterated greedy (TIG) algorithm to address the problem. In this study, a novel reconstruction mechanism is developed to explore the most valuable search space to save the evaluation efforts. The proposed TIG shows a superior performance by experimental results.

For minimizing the multi-objective FGSP and related problems, many researchers have also carried out in-depth research. Considering the hybrid flexible flowshop, Karimi et al. (2010) presented a multi-phase method to minimize the makespan and total weighted tardiness, simultaneously. Zandieh and Karimi (2011) developed a multi-population GA to search the Pareto optimal solution to minimize the makespan and total weighted tardiness. To increase the customers' satisfaction and decrease the producer's cost, Bozorgirad and Logendran (2013) minimized the total weighted tardiness, the work-in-process (WIP) inventory, and the total weighted completion time of the hybrid FGSP with parallel machines. Notably, all the machines and jobs may not be available at time 0. The MILP model of the hybrid FGSP is also designed for small-size problems. Finally, four efficient methods based on TS are designed for solving these problems. Qin et al. (2016) researched the FSP with learning and group effects based on position-dependent and optimized four objectives: Total completion time, total weighted completion time, maximum lateness, and makespan. Feng et al. (2018) considered the imperfect PM in the flexible FGSP and developed a machine-level model and a system-level model to record the machine reliability evolution and to obtain the plan of PM, respectively. Then, for minimizing the weighted earliness and tardiness of the FGSP, Keshavarz et al. (2019) presented an MILP model and designed a hybrid algorithm based on the PSO algorithm with timing and LB method to solve the FGSP. He et al. (2021) utilized a greedy CCEA with a greedy energy-saving strategy and random mutation operator to optimize the makespan, total flow time, and total energy consumption, simultaneously. The above research indicates that with the increase of optimization objectives, the problems are becoming increasingly complex. Designing an effective evolutionary optimization algorithm can effectively solve such problems (Tavakkoli-Moghaddam et al., 2010; Taghavi-fard et al., 2011; Gholipour-Kanani et al., 2011; Lu and Logendran, 2013).

Apart from the above constraints considered in GSP, blocking, unrelated parallel machines, skilled workforce assignment, failure rate threshold, due windows, and limited buffers are also considered in GSP (Logendran and Sriskandarajah, 1993; Behnamian et al., 2010; Solimanpur and Elmi, 2011; Costa et al., 2014; Neufeld et al., 2015). Zheng et al. (2014) assumed that the buffers between adjacent machines are limited and established a mathematical model of an FGSP with limited buffers. In this work, a hybrid differential evolution algorithm by combining differential evolution with TS is designed to minimize the total flow time. When the number of buffers is zero in the FSP, the problem becomes a blocking flowshop scheduling one. Yuan et al. (2020b) researched a two-stage FGSP with a special blocking constraint. Based on the blocking feature, the authors established an MILP model with makespan and proposed a co-evolutionary estimation of distributed algorithm. For the unrelated parallel machines in hybrid flowshop GSP (HFGSP), the coding and decoding of solutions are critical. In Yuan et al. (2022), job scheduling within each group and assignment of jobs at each stage were uniformly coded. A decoding method was designed by utilizing the strategies of load balancing and improved first-come-first-served methods. Song et al. (2020) studied the group scheduling of optimal setup uncorrelated parallel machine based on a genetic TS algorithm. Owing to natural decay and machining wear, the equipment cannot always be in good processing condition. The failure rate of machines should be considered in group production scheduling model. Liao et al. (2017) presented a hybrid maintenance strategy that combined PM and minimal repair, in which failure rate threshold and unexpected failure were considered to perform PM and minimal repair, respectively, based on machine usage. In addition, for the dynamic GSPs, some literature proposed robust metaheuristics.

In our review, most papers have been identified focusing on the building models and solutions to the FGSPs. Table 2 summarizes these papers. The above literature analysis indicates that the optimization objectives are mainly makespan, the total flow time, and the total weighted tardiness. However, in practical production, we do not only consider economic indicators, such as makespan, tardiness time, or earliness time, but also consider environmental protection, energy consumption, or energy consumption cost indicators from a sustainable manufacturing point of view. Thus, expending much effort to optimize the energy costs and reduce the energy consumption costs by reasonably scheduling sequence is necessary.

Table 2 Flowshop group scheduling problems

Reference	Objective		Algorithm		Model
	Single objective	Multi-objective	Heuristic	Meta-heuristic	
Baker (1990)	Makespan		Johnson's rule		√
Schaller (2001)	Makespan			Branch-and-bound algorithm	√
Yang (2002)	Makespan			Branch-and-bound algorithm	
Zandieh et al. (2009)		Makespan and CPU time		SA and GA	
Salmasi et al. (2010)	Total flow time			HACO	√
Karimi et al. (2010)		Makespan and total weighted tardiness		Multi-phase GA	√
Hajinejad et al. (2011)	Total flow time			PSO	
Salmasi et al. (2011)	Makespan			HACO	√
Zandieh and Karimi (2011)		Makespan and total weighted tardiness		Multi-population GA	√
Bozorgirad and Logendran (2013)		Total weighted completion time and total weighted tardiness		TS	√
Mendes et al. (2013)	Total flow time			VND and ILS	
Keshavarz and Salmasi (2014)	Total completion time			Hybrid GA	√
Keshavarz et al. (2015a)	Total completion time			Memetic algorithm	√
Khamseh et al. (2015)	Makespan			SA and GA	
Zandieh and Hashemi (2015)	The expected average of completion times			GA	
Neufeld et al. (2015)	Makespan		Nawaz–Enscore–Ham (NEH)		√
Qin et al. (2016)		M&TCT, total weighted completion time, maximum lateness		GA and quantum differential evolutionary algorithm	√
Adressi et al. (2016)	Maximum completion time			SA and GA	
Costa et al. (2017)	Makespan			Hybrid GA	
Behjat and Salmasi (2017)		Total completion time	NEH, SPT	PSO and variable neighborhood search	√
Feng et al. (2018)		PM cost, repair cost, job tardiness cost		SA embedded GA	√
Rossit et al. (2018)		Makespan, economic objective, due date		Hybrid algorithm	
Keshavarz et al. (2019)	Total weighted earliness and tardiness			PSO	√
Feng et al. (2019)	Makespan			GA	√
Liu (2020)	Total flow time			Constructive heuristics	
Costa et al. (2020)	Makespan			Parallel self-adaptive GA	√
Yuan et al. (2020a)	Makespan			CGA	√
Yuan et al. (2021)	Makespan			CDDEA	√
Wang et al. (2022)	Total tardiness		Earliest due date (EDD), smallest overall slack time (OSL)	TIG algorithm	√

2.3 Intelligent optimization algorithms for solving discrete scheduling problems

Many intelligent optimization algorithms are used for solving discrete scheduling problems. In addition to the aforementioned intelligent optimization algorithms, such as artificial bee colony (ABC), GA, SA, ant colony optimization (ACO), PSO, memetic algorithm, and differential evolution algorithm, some more recent studies about intelligent optimization algorithms for solving discrete scheduling problems have been published. These algorithms can help readers refer and learn, and design or propose a new hybrid algorithm for solving GSPs.

For distributed FSPs, Zhao et al. (2022) proposed a constructive heuristic and a water wave optimization algorithm with problem-specific knowledge, in which four local search methods of the variable neighborhood search strategy were presented to improve the performance of the algorithm. Yang and Li (2022) employed a knowledge-driven constructive heuristic to minimize the maximum assembly completion time, in which three different types of neighborhood knowledge were improved. Shao et al. (2021) proposed an efficient iterated greedy (IG) algorithm in which an improved NEH heuristic and a problem-specific knowledge-based destruction-construction were used to explore the solution space. Hamzadayı (2020) developed an effective benders decomposition algorithm that consists of the hybridization of NEH2_en and local search algorithm. Tang et al. (2022a) also adopted a hybrid teaching-learning-based optimization algorithm to solve the distributed sand-casting job shop scheduling problem. This algorithm used a TS based on the critical path to increase the number of teachers in the dynamic teacher group.

Population-based intelligent optimization algorithms have good diversity and are used to solve discrete scheduling problems. Guo et al. (2021) designed a bi-population immune algorithm to optimize the weapon transportation support scheduling problem. In this work, a population-based forward/backward scheduling technique, local search strategy, and a chaotic catastrophe operator are embedded. Fernandez-Viagas and Costa (2021) proposed a population-based constructive heuristic based on a beam search strategy to solve the single-machine scheduling problem with sequence-dependent setup times and release times. Chen et al. (2021) developed a population perturbation and elimination strategy-based GA to optimize the multi-satellite tracking telemetry and command scheduling problem.

The RL method has been used to intelligently adjust some parameters or design selection strategy in intelligent optimization algorithms. Chen et al. (2020) proposed a self-learning GA integrated RL to solve the flexible job shop scheduling problem. In this algorithm, a Q-learning algorithm is considered the learning method at a later evolutionary stage. Koksal et al. (2021) adopted three RL-enabled evolutionary algorithms, i.e., RL-enabled GA, RL-enabled ACO, and RL-enabled PSO, to optimize the integrated school bus routing and scheduling problem. Under the RL framework, Ren et al. (2021) mapped the various information of the FSP and the optimal scheduling rules into states and actions of RL, respectively. The authors trained a neural network to establish the mapping between states and actions. For the distributed assembly no-idle FSP, Zhao et al. (2021) designed an RL mechanism in the propagation phase of water wave optimization algorithm to balance the exploration and exploitation of the proposed algorithm. Recently, Chen et al. (2022) developed a distributed RL algorithm to solve a multi-wave firefighting scheduling problem. In this proposed algorithm, a local Q-function is designed to find the optimal solution.

Estimation of distribution algorithm (EDA) is also applied to solve the discrete scheduling problem and shows superiority. Wu et al. (2021) proposed a path relinking to enhance EDA to address the direct acyclic graph task scheduling problem. They considered path relinking-based knowledge as a local search strategy. To improve the performance of EDA, Zhang et al. (2022b) designed a novel matrix-cube-based EDA to optimize the makespan criterion of the blocking FSP with sequence-dependent setup times. Zhou et al. (2021) employed an improved EDA including a ranking and selection method to perform project scheduling. The experiment results demonstrated that the improved EDA obtains 23% higher expected makespan values for practical cases.

For the multi-objective flowshop scheduling optimization problems, Zhang et al. (2022a) proposed an automatic multi-objective evolutionary algorithm. In this algorithm, an automated algorithm design philosophy is used. Wu and Cao (2022) considered the energy consumption of a re-entrant hybrid FSP (HFSP) and developed an improved multi-objective evolutionary algorithm based on decomposition. For the multi-objective distributed assembly permutation FSP, Huang et al. (2022) employed a two-phase evolutionary algorithm. The first phase adopts a two-population structure to minimize the two objectives. The second phase adopts two new crossover operators to improve performance of the proposed algorithm. Considering the device dynamic reconfiguration processes, Wang et al. (2021) simultaneously optimized the makespan and the whole device's energy consumption and proposed a multi-objective whale optimization algorithm to address the above problem.

3 Notations, mathematical models, and existing benchmark instances

For optimizing the GSPs, the mathematical models of different types of GSP should be constructed. Specifically, the MILP should be considered one of the best mathematical models. Therefore, this paper formulated the common mathematical models of single-machine GSP, PFGSP, and distributed PFGSP (DPFGSP), respectively, according to the previous classifications. The minimal objective function of these problems is the makespan. The Gurobi optimizer has the capacity of finding optimal solutions within the acceptable CPU time for small problems. The following mathematical models can help the readers understand and solve the corresponding problems.

3.1 Notations

In this section, all notations and decision variables of the following three mathematical models are given. The optimization objective of all the mathematical models considered in this paper is minimizing maximal complete time (or makespan), i.e., Minimize $C_{\max}$.

Parameters:

- L Number of cellulars.
- l, l' Index of cellulars, $l, l' \in \{1, 2, ..., L\}$.
- M Number of machines in flowshop or in each cellular for DPFGSP.
- m Index of machines.
- F Number of families or groups.
- f, f' Index of families, $f, f' \in \{0, 1, ..., F\}$. 0 is the index of dummy family, which expresses the start and end of the family sequence on each machine. In this paper, dummy family is used in all models.
- J Number of jobs.
- j, j' Index of jobs, $j, j' \in \{0, 1, ..., J\}$. 0 is the index of dummy job, which expresses the beginning and finishing of the scheduling sequence in a family.
- ω_f Set of jobs in family f.
- J_f Number of jobs in family f, $\sum_{f=0}^{F} J_f = J$.
- r Index of positions of the family sequence, $r \in \{1, 2, ..., F\}$.
- k Index of positions of the job sequence in family f, $k \in \{1, 2, ..., J_f\}$.
- $pt_{j,m}$ Processing time of job j on machine m.
- $st_{f,f',m}$ Setup time from family f to f' on machine m, in which $st_{0,f',m}$ is an initial setup time when f' is assigned to the first position on machine m.
- $pt'_{f,j,k}$ Processing time of the kth scheduled job j in family f (single-machine GSP).
- $st'_{f,r}$ Setup time of the rth scheduled family f (single-machine GSP).
- h Sufficiently large positive number.

Decision variables:

- $C_{\max}$ Makespan of scheduling sequence.
- $u_{f,r}$ Binary decision variable, 1 represents that the family f is assigned to the rth position in the family sequence, 0 otherwise.
- $v_{f,j,k}$ Binary decision variable, 1 if the job j in family f is assigned to the kth position of the job sequence in family f, 0 otherwise.
- $c_{j,m}$ Completion time of job j on machine m.
- $x_{f,f'}$ Binary decision variable, 1 and 0 represent that the family f' is and is not a direct successor of f, respectively.
- $y_{j,j'}$ Binary decision variable, 1 and 0 represent that the job j' is and is not a direct successor of the job j which belongs to the same family with j', respectively.
- $w_{f,l}$ Binary decision variable, 1 and 0 represent that the family f is and is not allocated to cellular l, respectively.
- $u'_{f,l,r}$ Binary decision variable, 1 represents that the family f is the rth scheduled family in cellular l, 0 otherwise.

3.2 Single-machine group scheduling problem

The basic assumptions about the single-machine GSPs are listed as follows:

(1) There are families processed on the machine. Each family consists of multiple jobs, and each job can be processed on the machine for one time.

(2) Family preemption is not permitted.

(3) Setup time is sequence-dependent.

(4) The jobs in the same family are not separated and mixed with the jobs belonging to other families.

(5) The processing of a job is not allowed to be interrupted.

The mathematical model of single-machine GSP is subjected to constraint sets (1)–(5).

$$\sum_{f=1}^{F} u_{f,r} = 1, \forall r \in \{1, 2, ..., F\}, \quad (1)$$

$$\sum_{r=1}^{F} u_{f,r} = 1, \forall f \in \{1, 2, ..., F\}, \quad (2)$$

$$\sum_{j \in \omega_f} v_{f,j,k} = 1, \forall f \in \{1, 2, ..., F\}, \forall k \in \{1, 2, ..., J_f\}, \quad (3)$$

$$\sum_{k=1}^{J_f} v_{f,j,k} = 1, \forall f \in \{1, 2, ..., F\}, \forall j \in \{1, 2, ..., J_f\}, \quad (4)$$

$$C_{\max} = \sum_{f=1}^{F} \sum_{r=1}^{F} st'_{f,r} u_{f,r} + \sum_{f=1}^{F} \sum_{j \in \omega_f} \sum_{k=1}^{J_f} pt'_{f,j,k} v_{f,j,k}. \quad (5)$$

Constraint (1) represents that any position of the family sequence can be assigned to no more than one family simultaneously. Constraint (2) guarantees that any family can be allocated to a location of the family sequence simultaneously. Constraint (3) represents that any location of the job sequence in family f can be assigned to no more than one job from family f simultaneously. Constraint (4) guarantees that any job of family f is allocated to one location of the job sequence in family f simultaneously. In addition, constraints (3) and (4) determine that the jobs of the same family are not divided and mixed with the ones belonging to other families. Constraint (5) calculates makespan.

The following is a specific example of single-machine GSP. Let $F = 4$, $J = 8$, $\omega_1 = \{1, 2\}$, $\omega_2 = \{3, 4\}$, $\omega_3 = \{5\}$, and $\omega_4 = \{6, 7, 8\}$. Table 3 shows the processing times of jobs at different positions and Table 4 shows the setup times of groups at different positions.

We adopt the Gurobi solver to obtain an optimal solution for this example, i.e., the group sequence is (1, 4, 3, 2), and the job sequence in each group is (2, 1), (6, 7, 8), (5), and (4, 3), respectively. The Gantt chart of the optimal solution is given in Fig. 4.

3.3 Permutation flowshop group scheduling problem

The following gives the basic assumptions of PFGSP:

(1) A set of J jobs are grouped into F families and processed on M machines without preemption.

(2) Each job is uninterruptedly processed on one machine at most, and each machine processes one job at most.

Table 3 Processing times of jobs at different positions

Group	Job	Position in group		Group	Job	Position in group		
		1	2			1	2	3
ω_1	1	4	3	ω_3	5	2		
	2	5	6	ω_4	6	3	4	2
ω_2	3	3	2		7	4	2	3
	4	2	4		8	4	5	1

Table 4 The setup times of groups at different positions

Group	Position on machine			
	1	2	3	4
ω_1	3	3	2	4
ω_2	5	7	2	1
ω_3	5	3	3	5
ω_4	2	1	4	3

(3) The storage between the adjacent machines is unlimited.

(4) Family preemption is not allowed, and neither are job preemption nor part preemption.

(5) The setup times of each family on the machine only rely on its direct preceding family.

(6) The setup times of each job from the same family is not required.

(7) The jobs belonging to the same family are not divided and mixed with the jobs in other families.

To comprehensively understand PFGSP, this paper constructs two types of mathematical models based on adjacent and position-based modeling methods, respectively.

3.3.1 Mathematical models based on adjacent modeling method

The mathematical model of PFGSP based on the adjacent modeling method is subjected to constraint sets (6)–(14).

$$\sum_{f'=0, f'\neq f}^{F} x_{f,f'} = 1, \ \forall f \in \{1, 2, ..., F\}, \quad (6)$$

$$\sum_{f=0, f\neq f'}^{F} x_{f,f'} = 1, \ \forall f' \in \{1, 2, ..., F\}, \quad (7)$$

$$\sum_{j'\in\{0\}\cup\omega_f,\, j'\neq j} y_{j,j'} = 1, \ \forall f \in \{1, 2, ..., F\}, \ \forall j \in \{0\}\cup\omega_f, \quad (8)$$

$$\sum_{j\in\{0\}\cup\omega_f,\, j\neq j'} y_{j,j'} = 1, \ \forall f \in \{1, 2, ..., F\}, \ \forall j' \in \{0\}\cup\omega_f, \quad (9)$$

$$c_{j',m} \geqslant c_{j,m} + pt_{j',m} + (y_{j,j'} - 1)\cdot h, \ \forall f \in \{1, 2, ..., F\}, \ \forall j, \ j' \in \omega_f, \ j' \neq j, \ \forall m \in \{1, 2, ..., M\}, \quad (10)$$

$$c_{j',m} \geqslant c_{j,m} + st_{f,f',m} + pt_{j',m} + (x_{f,f'} - 1)\cdot h, \ \forall f, \ f' \in \{1, 2, ..., F\}, \ f \neq f', \ \forall j \in \omega_f, \ \forall j' \in \omega_{f'}, \ \forall m \in \{1, 2, ..., M\}, \quad (11)$$

$$c_{j,m} \geqslant st_{0,f',m} + pt_{j,m} + (x_{0,f'} - 1)\cdot h, \ \forall f' \in \{1, 2, ..., F\}, \ \forall j \in \omega_{f'}, \ \forall m \in \{1, 2, ..., M\}, \quad (12)$$

$$c_{j,m+1} \geqslant c_{j,m} + pt_{j,m+1}, \ \forall j \in \{1, 2, ..., J\}, \ \forall m \in \{1, 2, ..., M-1\}, \quad (13)$$

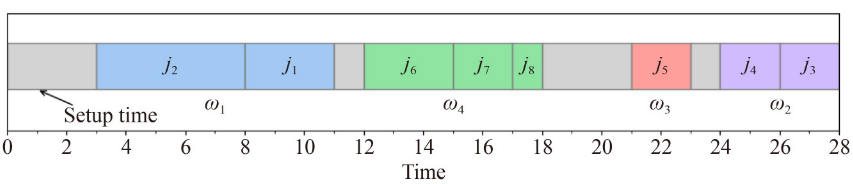

Fig. 4 Gantt chart for the optimal solution to the example problem of single-machine GSP.

$$C_{\max} \geqslant c_{j,M},\ \forall j \in \{1, 2, ..., J\}. \tag{14}$$

The model adopts sequence-based variables with two dummy families and two dummy jobs in each family. The family scheduling sequence begins from a dummy family and finishes up with another one. The job scheduling sequence in each family begins with a dummy job and finishes up with another one.

Constraints (6) and (7) ensure that each family, including two dummy families, has only one direct predecessor and direct successor in the family sequence. Constraints (8) and (9) enforce that each job in the same family f, including two dummy jobs, has only one direct successor and predecessor. In addition, constraints (8) and (9) determine that the jobs from the same family are not separated and mixed with the jobs belonging to other families. In constraint (10), for job j and its direct successor j' on machine m, when j and j' come from the same family, $c_{j',m}$ is greater than or equal to that of j plus $pt_{j',m}$. Otherwise, when j and j' come from two different families f and f', respectively, $st_{f,f',m}$ should be added to $c_{j',m}$, as shown in constraint (11). When the jobs belonging to the first family are processed on machine m, $st_{0,f',m}$ is considered by constraint (12). In addition, constraints (10), (11) and (12) eliminate subtours. Constraint (13) defines that the completion time of a job in a machine is larger than or equal to the processing time in the same machine plus its completion time in the previous machine. The makespan is defined by constraint (14).

3.3.2 Mathematical models based on position-based modeling method

The mathematical model of PFGSP based on the position-based modeling method is subjected to constraint sets (15)–(23).

$$\sum_{f=1}^{F} u_{f,r} = 1,\ \forall r \in \{1, 2, ..., F\}, \tag{15}$$

$$\sum_{r=1}^{F} u_{f,r} = 1,\ \forall f \in \{1, 2, ..., F\}, \tag{16}$$

$$\sum_{j\in\omega_f} v_{f,j,k} = 1,\ \forall f \in \{1, 2, ..., F\},\ \forall k \in \{1, 2, ..., J_f\}, \tag{17}$$

$$\sum_{k=1}^{J_f} v_{f,j,k} = 1,\ \forall f \in \{1, 2, ..., F\},\ \forall j \in \{1, 2, ..., J_f\}, \tag{18}$$

$$\begin{gathered} c_{j',m} \geqslant c_{j,m} + pt_{j',m} + (v_{f,j,k} + v_{f,j',k+1} - 2)\cdot h,\ \forall f \in \{1, 2, ..., F\}, \\ \forall j \in \omega_f, \forall j' \in \omega_{f'},\ \forall m \in \{1, 2, ..., M\}, \\ \forall k \in \{1, 2, ..., J_f - 1\}, \end{gathered} \tag{19}$$

$$\begin{gathered} c_{j',m} \geqslant c_{j,m} + st_{f,f',m} + pt_{j',m} + \left(u_{f,r} + u_{f',r+1} + v_{f,j,J_f} + v_{f',j',1} - 4\right)\cdot h, \\ \forall f,\ f' \in \{1, 2, ..., F\},\ f \neq f',\ \forall j \in \omega_f,\ \forall j' \in \omega_{f'}, \\ \forall m \in \{1, 2, ..., M\},\ \forall r \in \{1, 2, ..., F-1\}, \end{gathered} \tag{20}$$

$$\begin{gathered} c_{j,m} \geqslant st_{0,f',m} + pt_{j,m} + (u_{f',1} + v_{f',j,1} - 2)\cdot h, \\ \forall f' \in \{1, 2, ..., F\},\ \forall j \in \omega_{f'}, \\ \forall m \in \{1, 2, ..., M\}, \end{gathered} \tag{21}$$

$$\begin{gathered} c_{j,m+1} \geqslant c_{j,m} + pt_{j,m+1},\ \forall j \in \{1, 2, ..., J\}, \\ \forall m \in \{1, 2, ..., M-1\}, \end{gathered} \tag{22}$$

$$C_{\max} \geqslant c_{j,M},\ \forall j \in \{1, 2, ..., J\}. \tag{23}$$

Constraint (15) shows that any position of the family sequence can be assigned one family simultaneously. Constraint (16) guarantees that any family can be allocated to a position of the family sequence simultaneously. Constraint (17) represents that any position of the job sequence in family f can be assigned one job from family f simultaneously. Constraint (18) guarantees that any job from family f can be allocated to a position of the job sequence in family f simultaneously. In addition, constraints (17) and (18) determine that the jobs in the same family are not divided and mixed with the jobs belonging to other families. For jobs j and j' that are from the same family f on machine m, if they are put into the kth and $(k+1)$th position of the job sequence in family f, respectively, $c_{j',m}$ is not less than $c_{j,m}$ plus $pt_{j',m}$, ensured by constraint (19). In constraint (20), for families f and f' on machine m, if they are put into the rth and $(r+1)$th position of the family sequence, respectively, the completion time of the last job j in the family f on machine m is not less than the completion time of the first job j' in the family f' on machine m plus $pt_{j',m}$ and family setup time $st_{f,f',m}$. When the jobs belonging to the first family are processed on machine m, $st_{0,f',m}$ is considered by constraint (21). In addition, constraints (19), (20), and (21) eliminate subtours. Constraint (22) defines that the completion time of a job in a machine is larger than or equal to the processing time in the same machine plus its completion time in the previous machine. The makespan is defined by constraint (23).

We consider an example of PFGSP with $F = 4$, $M = 2$, $J = 8$, $\omega_1 = \{1, 2\}$, $\omega_2 = \{3, 4\}$, $\omega_3 = \{5\}$, and $\omega_4 = \{6, 7, 8\}$. The processing times and the family setup times are shown in Tables 5 and 6, respectively.

We adopt the Gurobi solver to obtain an optimal solution for the above example, i.e., the group sequence is (4, 1, 3, 2), and the job sequence in each group is (6, 8, 7), (2, 1), (5), and (4, 3), respectively. The Gantt chart of the optimal solution is given in Fig. 5.

Table 5 Processing times of jobs on machines M_1 and M_2

Group	Job	Machine	
		M_1	M_2
ω_1	1	4	4
	2	2	3
ω_2	3	3	2
	4	4	4
ω_3	5	2	3
ω_4	6	3	4
	7	4	2
	8	4	5

Table 6 Setup times of groups on machines M_1 and M_2

Group	M_1				M_2			
	ω_1	ω_2	ω_3	ω_4	ω_1	ω_2	ω_3	ω_4
ω_0	4	4	3	2	5	4	3	4
ω_1	-	5	2	2	-	7	1	4
ω_2	5	-	5	3	4	-	3	5
ω_3	2	2	-	4	2	1	-	3
ω_4	2	3	3	-	3	4	2	-

3.4 Distributed permutation flowshop group scheduling problem

The definition of the basic assumptions about the DPFGSP is listed as follows:

(1) There is a set of L same factories or cellulars, and each factory or cellular has a flowshop technological process with M machines.

(2) All J jobs are grouped into F families, each of which can be processed at any of the cellulars.

(3) The storage between the adjacent machines is unlimited.

(4) Family preemption is not allowed, and neither are job preemption nor part preemption.

(5) The setup time of each family only relies on its direct preceding family.

(6) The setup time of each job from one family is not needed.

(7) A job can be uninterruptedly processed on at most one machine, and a machine can process most one job at a time.

(8) The jobs from the same family are not divided and mixed with the jobs belonging to other families.

To comprehensively understand DPFGSP, this paper also constructs two types of mathematical models based on adjacent and position-based modeling methods, respectively.

3.4.1 Mathematical models based on adjacent modeling method

The mathematical model of DPFGSP based on the adjacent modeling method is subjected to constraint sets (24)–(35).

$$\sum_{f'=0, f'\neq f}^{F} x_{f,f'} = 1, \forall f \in \{1, 2, ..., F\}, \tag{24}$$

$$\sum_{f=0, f\neq f'}^{F} x_{f,f'} = 1, \forall f' \in \{1, 2, ..., F\}, \tag{25}$$

$$\sum_{f'=1}^{F} x_{0,f'} \leqslant L, \tag{26}$$

$$\sum_{f=1}^{F} x_{f,0} \leqslant L, \tag{27}$$

$$\sum_{f'=1}^{F} x_{0,f'} = \sum_{f=1}^{F} x_{f,0}, \tag{28}$$

$$\sum_{j'\in\{0\}\cup\omega_f,\, j'\neq j} y_{j,j'} = 1, \forall f \in \{1, 2, ..., F\}, \forall j \in \{0\}\cup\omega_f, \tag{29}$$

$$\sum_{j\in\{0\}\cup\omega_f,\, j\neq j'} y_{j,j'} = 1, \forall f \in \{1, 2, ..., F\}, \forall j' \in \{0\}\cup\omega_f, \tag{30}$$

$$c_{j',m} \geqslant c_{j,m} + pt_{j',m} + (y_{j,j'} - 1)\cdot h, \forall f \in \{1, 2, ..., F\}, \forall j \in \omega_f, \forall j' \in \omega_{f'}, \forall m \in \{1, 2, ..., M\}, \tag{31}$$

$$c_{j',m} \geqslant c_{j,m} + st_{f,f',m} + pt_{j',m} + (x_{f,f'} - 1)\cdot h, \forall f, f' \in \{1, 2, ..., F\}, f \neq f', \forall j \in \omega_f, \forall j' \in \omega_{f'}, \forall m \in \{1, 2, ..., M\}, \tag{32}$$

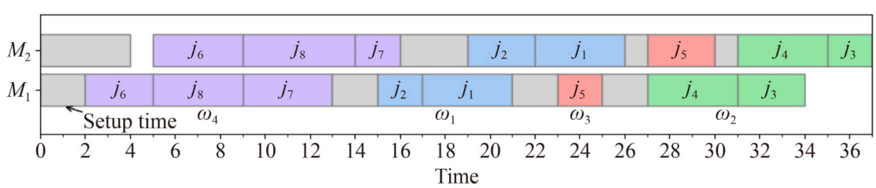

Fig. 5 Gantt chart for the optimal solution to the example problem of PFGSP.

$$c_{j,m} \geqslant st_{0,f',m} + pt_{j,m} + (x_{0,f'} - 1) \cdot h, \ \forall f' \in \{1, 2, ..., F\}, \ \forall j \in \omega_{f'}, \ \forall m \in \{1, 2, ..., M\}, \tag{33}$$

$$c_{j,m+1} \geqslant c_{j,m} + pt_{j,m+1}, \ \forall j \in \{1, 2, ..., J\}, \ \forall m \in \{1, 2, ..., M-1\}, \tag{34}$$

$$C_{\max} \geqslant c_{j,M}, \ \forall j \in \{1, 2, ..., J\}. \tag{35}$$

The above model adopts sequence-based variables with less than or equal to L dummy families and two dummy jobs in each family. The family begins with a dummy family and finishes up with another dummy one. The remaining dummy families divide the family sequence into some subsequences. Each subsequence is for one cellular. The job sequence in each family begins with a dummy job and finishes up with another dummy one.

Constraints (24) and (25) ensure that each family has only one direct successor and predecessor in the family sequence. Constraints (26) and (27) guarantee that the dummy family is a direct predecessor and successor of the families less than or equal to L times, separately. Constraint (28) ensures that the dummy family has the same number of direct predecessors and successors in the family sequence. Constraints (29) and (30) enforce that each job in the family f, including two dummy jobs, have only one direct successor and predecessor. In addition, in constraints (29) and (30), the jobs in the same family are not divided and mixed with the jobs belonging to other families. In constraint (31), for job j and its direct successor j' on machine m, when j and j' come from the same family, $c_{j',m}$ is not less than $c_{j,m}$ plus $pt_{j',m}$. Otherwise, when j and j' come from different families f and f', respectively, $c_{j',m}$ should add $st_{f,f',m}$, as shown in constraint (32). When the jobs belonging to the first family are processed on machine m, $st_{0,f',m}$ is considered by constraint (33). In addition, constraints (31), (32), and (33) eliminate subtours. Constraint (34) defines that the completion time of a job in a machine is larger than or equal to the processing time in the same machine plus its completion time in the previous machine. The makespan is defined by constraint (35).

3.4.2 Mathematical models based on position-based modeling method

The mathematical model of DPFGSP based on the position-based modeling method is subjected to constraint sets (36)–(46).

$$\sum_{l=1}^{L} w_{f,l} = 1, \ \forall f \in \{1, 2, ..., F\}, \tag{36}$$

$$\sum_{f=1}^{F} u'_{f,l,r} \leqslant 1, \ \forall l \in \{1, 2, ..., L\}, \ \forall r \in \{1, 2, ..., F\}, \tag{37}$$

$$\sum_{r=1}^{F} u'_{f,l,r} = w_{f,l}, \ \forall f \in \{1, 2, ..., F\}, \ \forall l \in \{1, 2, ..., L\}, \tag{38}$$

$$\sum_{f=1}^{F} u'_{f,l,r} \geqslant \sum_{f=1}^{F} u'_{f,l,r+1}, \ \forall l \in \{1, 2, ..., L\}, \ \forall r \in \{1, 2, ..., F-1\}, \tag{39}$$

$$\sum_{j \in \omega_f} v_{f,j,k} = 1, \ \forall f \in \{1, 2, ..., F\}, \ \forall k \in \{1, 2, ..., J_f\}, \tag{40}$$

$$\sum_{k=1}^{J_f} v_{f,j,k} = 1, \ \forall f \in \{1, 2, ..., F\}, \ \forall j \in \{1, 2, ..., J_f\}, \tag{41}$$

$$c_{j',m} \geqslant c_{j,m} + pt_{j',m} + (v_{f,j,k} + v_{f,j',k+1} - 2) \cdot h, \ \forall f \in \{1, 2, ..., F\}, \ \forall j \in \omega_f, \ \forall j' \in \omega_{f'}, \ \forall m \in \{1, 2, ..., M\}, \ \forall k \in \{1, 2, ..., J_f - 1\}, \tag{42}$$

$$c_{j',m} \geqslant c_{j,m} + st_{f,f',m} + pt_{j',m} + \left(u'_{f,l,r} + u'_{f',l,r+1} + v_{f,j,J_f} + v_{f',j',1} - 4\right) \cdot h, \ \forall f, f' \in \{1, 2, ..., F\}, \ f \neq f', \ \forall l \in \{1, 2, ..., L\}, \ \forall j \in \omega_f, \ \forall j' \in \omega_{f'}, \ \forall m \in \{1, 2, ..., M\}, \ \forall r \in \{1, 2, ..., F-1\}, \tag{43}$$

$$c_{j,m} \geqslant st_{0,f',m} + pt_{j,m} + \left(u'_{f',l,1} + v_{f',j,1} - 2\right) \cdot h, \ \forall f' \in \{1, 2, ..., F\}, \ \forall l \in \{1, 2, ..., L\}, \ \forall j \in \omega_{f'}, \ \forall m \in \{1, 2, ..., M\}, \tag{44}$$

$$c_{j,m+1} \geqslant c_{j,m} + pt_{j,m+1}, \ \forall j \in \{1, 2, ..., J\}, \ \forall m \in \{1, 2, ..., M-1\}, \tag{45}$$

$$C_{\max} \geqslant c_{j,M}, \ \forall j \in \{1, 2, ..., J\}. \tag{46}$$

Constraint (36) is that each family is allocated exactly to a cellular. Constraint (37) represents that any position of the family sequence can be assigned no more than one family in a cellular simultaneously. Constraint (38) enforces that any family assigned to the cellular l can be put into one position of the family sequence in the cellular l simultaneously. Constraint (39) guarantees that families must be put into the positions of the family sequence in sequential order. Constraint (40) represents that any position of the job sequence in family f can be assigned no more than one job from family f simultaneously. Constraint (41) guarantees that any job from family f is put into one position of the job sequence in family f simultaneously. In addition, constraints (40) and (41) determine that the jobs belonging to the same family are not divided and mixed with the jobs from other families. For jobs j and j' from the same family f on machine m, if they are put into the kth and $(k+1)$th position of the job sequence in family f, respectively, $c_{j',m}$ is not less than $c_{j,m}$ plus $pt_{j',m}$, ensured by constraint (42). For families f and f' on machine m in cellular l, if they are put into the rth and $(r+1)$th position of the family sequence, respectively, the completion time of the last job j in the family f on machine m is not less than the completion time of the first job j' in the family f' on machine m plus $pt_{j',m}$ and $st_{f,f',m}$, as shown in constraint (43). When the first job in the first family is processed on machine m in cellular l, $st_{0,f',m}$ is considered by constraint (44). In addition, constraints (42), (43), and (44) eliminate subtours. Constraint (45) defines that the completion time of a job in a machine is larger than or equal to the processing time in the same machine plus its completion time in the previous machine. The makespan is defined by constraint (46).

On the basis of the previous example of PFGSP, we let $L=2$, which gives us the example of the DPFGSP. We adopt the Gurobi solver to obtain an optimal solution for the example, i.e., the group sequences for the two factories are (2, 3) and (4, 1), respectively, and the job sequence in each group is (3, 4), (5), (6, 8, 7), and (1, 2). The Gantt chart of the optimal solution is given in Fig. 6.

3.5 Existing benchmark instance

To verify the property of the proposed algorithms, the benchmark instances of GSPs are necessary. Generally, three different types of scale instances are considered: Small, medium, and large. The numbers of jobs, machines, and groups are set according to the scale of the test instances, i.e., $n=\{8, 10, 20, 40, 60, 90, 100, 150\}$, $m=\{2, 4, 6, 10\}$, and $g=\{3, 4, 5, 10, 20, 50\}$. For the distributed GSPs, the number of factories or cellulars, L, is {2, 3, 4, 5, 6, 7}. These values can lead to different scale combinations. For each combination, several instances with the same scale and different values of processing time can be obtained. The processing times of jobs on a machine have been generated randomly in the range [1, 10] or [1, 20]. The setup times between families have been randomly drawn in the range [1, 20], [10, 50], or [1, 100]. For the ratio of mean group setup time to mean job processing time, Schaller et al. (2000) subdivided it into three ratios, i.e., 2:1, 5:1, and 10:1. These ratio values can be regarded as a factor when generating different test instances. However, for the cellular scheduling problems, a gap exists between the simulation data and the practical production data. Thus, we should continue to develop benchmark instances by considering different material flows and the model of real-world cellular manufacturing systems.

4 Research survey results

This paper analyzed the reviewed literature in terms of GSPs having single and multiple objective functions, solution algorithms used, and their mathematical models to clearly understand the existing research gaps and provide suggestions for researchers. A total of 109 papers on GSPs have been reviewed. The statistic proportions of papers with respect to the objective functions, the number of objective functions, solving algorithms, and the production environment are displayed in Fig. 7, respectively.

The makespan (completion time) is the most widely studied single-objective optimization function, followed by total flow time, tardiness time, earliness time, and others. As shown in Figs. 7(a) and 7(b), most papers studied a single objective of GSP (89%), in which 83% of papers optimized the makespan objective, followed by other objectives (i.e., costs and eliminating the role of the kitting staff, 8%), total flow time (7%), and total completion time (2%).

To optimize GSPs, some algorithms are proposed. Solving algorithms are categorized into four categories: Exact methods, heuristics, metaheuristics, and hybrid algorithms. Exact algorithms mainly include PTA, branch-and-bound, MILP, and bounded dynamic programming. The heuristics refer to SPT, LPT, NEH, and EDD. Metaheuristics are GA, IG, SA, PSO, VND, ACO, and EDA. Owing to the complex nature of GSPs, hybrid algorithms are often adopted by integrating local search strategies.

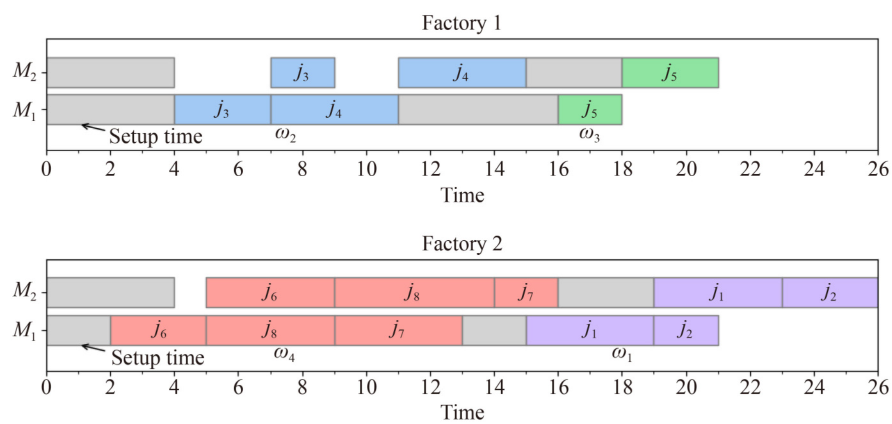

Fig. 6 Gantt chart for the optimal solution to the example problem of DPFGSP.

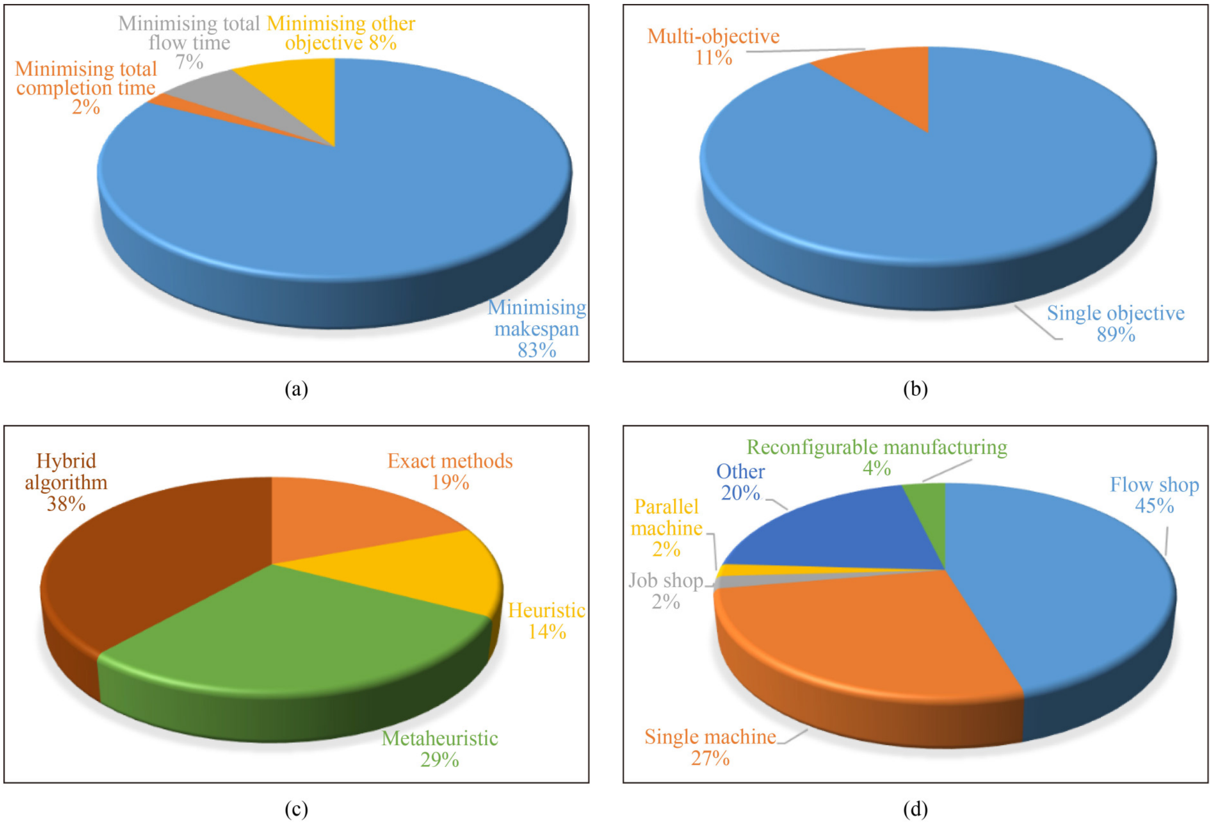

Fig. 7 Statistic proportion of papers based on (a) the objective functions; (b) the number of objective functions; (c) the solving methods; (d) the production environment.

The local search strategies are very important for the discrete GSPs. Most of the hybrid algorithms used local search to disturb the current solutions. The local search strategies as part of the hybrid algorithms have been effective to generate solutions with high quality in reasonable computational time. The static proportion of each category is reported in Fig. 7(c), in which the proportion of hybrid algorithms and metaheuristics is 67%. For large-scale test instances, the exact algorithms and heuristics cannot obtain an optimal solution within the acceptable CPU time.

As mentioned in Section 2, the scheduling problem in cellular manufacturing systems, called the cell scheduling problem, is generally modeled similar to the FGSP. However, the cell scheduling problem has some distinct characteristics, i.e., job shop cells, single-stage cells, parallel-machine cells, and flowshop cells. As shown in Fig. 7(d), 45% of the published papers considered the flowshop production environment, and 27% of the papers addressed a single machine. A total of 4% of the papers have addressed the RMS, and only 2% of the papers have addressed job shop and parallel machine production environment, respectively. The flowshop production technology exists extensively in cellular manufacturing and provides a simple and flexible production structure.

5 Conclusions and further research directions

Group scheduling is the key to the cellular manufacturing system and has attracted much concern. Owing to the production constraints and the large scale that exists extensively in cellular manufacturing, the existing scheduling models and algorithms are facing challenges. However, the emergence of the heuristic, intelligence optimization, and hybrid algorithms have resulted in breakthroughs and provided diversified optimization methods. In these ways, some feasible optimal solutions or scheduling schemes are obtained. The main contributions of this paper are as follows.

(1) This paper reviews and classifies 109 papers that deal with the GSPs in single and parallel machines, flowshop, RMSs, and job shop scheduling production environments. In general, completion time-based criteria are the most studied performance measures, being a target of 83% of the papers. Regarding the approaches or algorithms, 67% of the papers adopted hybrid algorithms and metaheuristics due to the complex nature of GSPs. In addition, this paper summarized the constraints, namely, blocking, unrelated parallel machines, skilled workforce assignment, failure rate threshold, due windows, and limited buffers, that can be considered in GSP.

(2) The classical mathematical models and optimization algorithms are reviewed. For optimizing the GSPs, the mathematical models of different types of GSP should be constructed. Specifically, MILP should be considered one of the best mathematical models. Thus, our paper formulates the common mathematical models of single-machine GSP, PFGSP, and DPFGSP, respectively. Furthermore, to comprehensively understand PFGSP and DPFGSP, our paper constructs two types of mathematical models based on adjacent and position-based modeling methods, respectively. The proposed mathematical models can help readers understand and solve the corresponding problems.

However, related research still has insufficiencies and needs to be further explored in problems and algorithm fields. Based on the analysis of reviewed literature, further research directions are as follows.

(1) Research directions of problems

The existing papers mainly do research on PFGSPs. However, the practice production environments have many constraints. First, due to a large number of enterprises gradually turning to multi-regional cooperation under the influence of globalization, distributed scheduling has become a trend (Han et al., 2022). In this context, the DPFGSP has attracted the attention of academics and becomes an urgent problem to be solved. Second, for some factories, more than one parallel machine can increase the productivity and flexibility of the scheduling process, called the HFSP (Qin et al., 2022b; 2022c). In this case, if the group technology is considered in HFSP, a new problem is proposed, namely the HFGSP (Qin et al., 2022a). Considering its practical significance, group technology should be considered in manufacturing. Apart from the above-mentioned distribution and parallel machine constraints, the no-idle, no-wait, blocking, and their combination should be considered to meet real-life production constraints. In addition, few research has been done on multi-objective optimization with economic objectives, energy consumption, machine load balance, and delaying the due date. These problems and their mathematical models (single and multi-objective) have not yet been well studied. Thus, constructing the mathematical models by considering these constraints and objectives is necessary.

(2) Research directions of problem algorithms

The existing algorithms can solve the GSPs for small, medium, and large instances. However, some issues or insufficiencies still remain. First, the algorithms lack the theoretical analysis to verify their property. Second, the time complexity of hybrid algorithms and metaheuristic is high for large instances. Third, different algorithms lack cooperation with each other. Based on the analysis, adding the theoretical analysis based on the characteristics of problems is necessary. The researcher can analyze the features of the GSPs with different constraints by using CPLEX or Gurobi and observe the scheduling scheme or solution allocation under the case of different constraints. For the second issue, to reduce the time complexity of the whole algorithm, one approach is to propose a quick evaluating method or calculation of the objective function (Pan et al., 2022). However, designing simple evolution operators, local search, and neighborhood search is necessary. To enhance the performance of the optimization algorithm, a hybrid algorithm should be considered by the combination of heuristic, swarm intelligence, IG algorithm, or RL, in which the cooperating mechanism, the associated knowledge, adaptively adjusted search operations, and parameter settings should be designed. In addition, for the multi-objective optimization algorithms, a good selection strategy and a good evaluation indicator are critical.

In short, as production technology requires, the need for group scheduling patterns will increase even more to save cost and time. Given the analyses in this paper, the practical production constraints, the corresponding mathematical models, and optimization algorithms will be considered in the future. In addition, dynamic or uncertain factors such as machine breakdowns, changing due dates, urgent insertion orders, and uncertain job processing times would be noteworthy. The problems and algorithms in group scheduling can be comprehensively developed and enhanced.

References

Adressi A, Tasouji Hassanpour S, Azizi V (2016). Solving group scheduling problem in no-wait flexible flowshop with random machine breakdown. Decision Science Letters, 5(1): 157–168

Allison J D (1990). Combining Petrov's heuristic and the CDS heuristic in group scheduling problems. Computers & Industrial Engineering, 19(1): 457–461

Bai J, Li Z R, Huang X (2012). Single-machine group scheduling with general deterioration and learning effects. Applied Mathematical Modelling, 36(3): 1267–1274

Baker K R (1990). Scheduling groups of jobs in the two-machine flow shop. Mathematical and Computer Modelling, 13(3): 29–36

Behjat S, Salmasi N (2017). Total completion time minimisation of no-wait flowshop group scheduling problem with sequence dependent setup times. European Journal of Industrial Engineering, 11(1): 22–48

Behnamian J, Zandieh M, Fatemi Ghomi S M T (2010). Due windows group scheduling using an effective hybrid optimization approach. International Journal of Advanced Manufacturing Technology, 46(5–8): 721–735

Biskup D (1999). Single-machine scheduling with learning considerations. European Journal of Operational Research, 115(1): 173–178

Bozorgirad M A, Logendran R (2013). Bi-criteria group scheduling in hybrid flowshops. International Journal of Production Economics, 145(2): 599–612

Bozorgirad M A, Logendran R (2016). A comparison of local search algorithms with population-based algorithms in hybrid flow shop scheduling problems with realistic characteristics. International Journal of Advanced Manufacturing Technology, 83(5): 1135–1151

Chen M, Wen J, Song Y J, Xing L, Chen Y (2021). A population perturbation and elimination strategy based genetic algorithm for multi-satellite TT&C scheduling problem. Swarm and Evolutionary Computation, 65: 100912

Chen R, Yang B, Li S, Wang S (2020). A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem. Computers & Industrial Engineering, 149: 106778

Chen X, Fu J, Zhou J, Li Y (2022). Distributed reinforcement learning algorithm for multi-wave fire fighting scheduling problem. IFAC-PapersOnLine, 55(3): 245–250

Cho K K, Ahn B H (2003). A hybrid genetic algorithm for group scheduling with sequence dependent group setup time. International Journal of Industrial Engineering: Theory, Applications and Practice, 10(4): 442–448

Costa A, Cappadonna F A, Fichera S (2014). Joint optimization of a flow-shop group scheduling with sequence dependent set-up times and skilled workforce assignment. International Journal of Production Research, 52(9): 2696–2728

Costa A, Cappadonna F A, Fichera S (2017). A hybrid genetic algorithm for minimizing makespan in a flow-shop sequence-dependent group scheduling problem. Journal of Intelligent Manufacturing, 28(6): 1269–1283

Costa A, Cappadonna F V, Fichera S (2020). Minimizing makespan in a flow shop sequence dependent group scheduling problem with blocking constraint. Engineering Applications of Artificial Intelligence, 89: 103413

Feng H, Tan C, Xia T, Pan E, Xi L (2019). Joint optimization of preventive maintenance and flexible flowshop sequence-dependent group scheduling considering multiple setups. Engineering Optimization, 51(9): 1529–1546

Feng H, Xi L, Xiao L, Xia T, Pan E (2018). Imperfect preventive maintenance optimization for flexible flowshop manufacturing cells considering sequence-dependent group scheduling. Reliability Engineering & System Safety, 176: 218–229

Fernandez-Viagas V, Costa A (2021). Two novel population based algorithms for the single machine scheduling problem with sequence dependent setup times and release times. Swarm and Evolutionary Computation, 63: 100869

Frazier G V (1996). An evaluation of group scheduling heuristics in a flow-line manufacturing cell. International Journal of Production Research, 34(4): 959–976

Gelogullari C A, Logendran R (2010). Group-scheduling problems in electronics manufacturing. Journal of Scheduling, 13(2): 177–202

Gholipour-Kanani Y, Tavakkoli-Moghaddam R, Khorrami A (2011). Solving a multi-criteria group scheduling problem for a cellular manufacturing system by scatter search. Journal of the Chinese Institute of Industrial Engineers, 28(3): 192–205

Gu X (2022). Modeling of reconfigurable manufacturing system architecture with geometric machines and in-stage gantries. Journal of Manufacturing Systems, 62: 102–113

Guo F, Han W, Su X, Liu Y, Cui R (2021). A bi-population immune algorithm for weapon transportation support scheduling problem with pickup and delivery on aircraft carrier deck. Defence Technology, in press, doi:10.1016/j.dt.2021.12.006

Hajinejad D, Salmasi N, Mokhtari R (2011). A fast hybrid particle swarm optimization algorithm for flow shop sequence dependent group scheduling problem. Scientia Iranica, 18(3): 759–764

Hamzadayı A (2020). An effective benders decomposition algorithm for solving the distributed permutation flowshop scheduling problem. Computers & Operations Research, 123: 105006

Han X, Han Y, Zhang B, Qin H, Li J, Liu Y, Gong D (2022). An effective iterative greedy algorithm for distributed blocking flowshop scheduling problem with balanced energy costs criterion. Applied Soft Computing, 129: 109502

He X, Pan Q, Gao L, Wang L, Suganthan P N (2021). A greedy cooperative co-evolution ARY algorithm with problem-specific knowledge for multi-objective flowshop group scheduling problems. IEEE Transactions on Evolutionary Computation, in press, doi: 10.1109/TEVC.2021.3115795

Huang X, Wang M Z (2014). Single machine group scheduling with time and position dependent processing times. Optimization Letters, 8(4): 1475–1485

Huang X, Wang M Z, Wang J B (2011). Single-machine group scheduling with both learning effects and deteriorating jobs. Computers & Industrial Engineering, 60(4): 750–754

Huang Y Y, Pan Q K, Gao L, Miao Z H, Peng C (2022). A two-phase evolutionary algorithm for multi-objective distributed assembly permutation flowshop scheduling problem. Swarm and Evolutionary Computation, 74: 101128

Janiak A, Kovalyov M Y (1995). Single machine group scheduling with ordered criteria. Annals of Operations Research, 57(1): 191–201

Jiang R, Chen Y, Guan Z, Zhou H (2013). Constraint satisfaction modeling and solving method for single machine group scheduling problem. China Mechanical Engineering, 24(12): 1642–1649 (in Chinese)

Karimi N, Zandieh M, Karamooz H R (2010). Bi-objective group scheduling in hybrid flexible flowshop: A multi-phase approach. Expert Systems with Applications, 37(6): 4024–4032

Karimi N, Zandieh M, Najafi A A (2011). Group scheduling in flexible flow shops: A hybridised approach of imperialist competitive algorithm and electromagnetic-like mechanism. International Journal of Production Research, 49(16): 4965–4977

Keshavarz T, Salmasi N (2013). Makespan minimisation in flexible flowshop sequence-dependent group scheduling problem. International Journal of Production Research, 51(20): 6182–6193

Keshavarz T, Salmasi N (2014). Efficient upper and lower bounding methods for flowshop sequence-dependent group scheduling problems. European Journal of Industrial Engineering, 8(3): 366–387

Keshavarz T, Salmasi N, Varmazyar M (2015a). Minimizing total completion time in the flexible flowshop sequence-dependent group scheduling problem. Annals of Operations Research, 226(1): 351–377

Keshavarz T, Salmasi N, Varmazyar M (2019). Flowshop sequence-dependent group scheduling with minimisation of weighted earliness and tardiness. European Journal of Industrial Engineering, 13(1): 54–80

Keshavarz T, Savelsbergh M, Salmasi N (2015b). A branch-and-bound algorithm for the single machine sequence-dependent group scheduling problem with earliness and tardiness penalties. Applied

Mathematical Modelling, 39(20): 6410–6424

Khamseh A, Jolai F, Babaei M (2015). Integrating sequence-dependent group scheduling problem and preventive maintenance in flexible flow shops. International Journal of Advanced Manufacturing Technology, 77(1–4): 173–185

Koksal E, Hegde A R, Pandiarajan H P, Veeravalli B (2021). Performance characterization of reinforcement learning-enabled evolutionary algorithms for integrated school bus routing and scheduling problem. International Journal of Cognitive Computing in Engineering, 2: 47–56

Kuo W H (2012). Single-machine group scheduling with time-dependent learning effect and position-based setup time learning effect. Annals of Operations Research, 196(1): 349–359

Lee W C, Wu C C (2009). A note on single-machine group scheduling problems with position-based learning effect. Applied Mathematical Modelling, 33(4): 2159–2163

Li S (1997). A hybrid two-stage flowshop with part family, batch production, major and minor set-ups. European Journal of Operational Research, 102(1): 142–156

Li S, Ng C T, Yuan J (2011). Group scheduling and due date assignment on a single machine. International Journal of Production Economics, 130(2): 230–235

Li W X, Zhao C L (2015). Single machine scheduling problem with multiple due windows assignment in a group technology. Journal of Applied Mathematics & Computing, 48(1–2): 477–494

Li X, Bayrak A E, Epureanu B I, Koren Y (2018). Real-time teaming of multiple reconfigurable manufacturing systems. CIRP Annals, 67(1): 437–440

Liao W, Zhang X, Jiang M (2017). Production scheduling model considering failure rate threshold for group production. Computer Integrated Manufacturing Systems, 23(3): 599–608 (in Chinese)

Lin H T, Liao C J (2003). A case study in a two-stage hybrid flow shop with setup time and dedicated machines. International Journal of Production Economics, 86(2): 133–143

Lin S W, Ying K C (2019). Makespan optimization in a no-wait flowline manufacturing cell with sequence-dependent family setup times. Computers & Industrial Engineering, 128: 1–7

Liou C D, Hsieh Y C (2015). A hybrid algorithm for the multi-stage flow shop group scheduling with sequence-dependent setup and transportation times. International Journal of Production Economics, 170: 258–267

Liou C D, Hsieh Y C, Chen Y Y (2013). A new encoding scheme-based hybrid algorithm for minimising two-machine flow-shop group scheduling problem. International Journal of Systems Science, 44(1): 77–93

Liou C D, Liu C H (2010). A novel encoding scheme of PSO for two-machine group scheduling. In: 1st International Conference on Swarm Intelligence: Advances in Swarm Intelligence. Beijing: Springer, 128–134

Liu P, Tang L, Zhou X (2010). Two-agent group scheduling with deteriorating jobs on a single machine. International Journal of Advanced Manufacturing Technology, 47(5–8): 657–664

Liu Y (2020). Effective heuristics to minimize total flowtime for distributed flowshop group scheduling problems. In: 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE). Harbin: IEEE, 708–711

Logendran R (1992). Group scheduling problem: Key to flexible manufacturing systems. Computers & Industrial Engineering, 23(1): 113–116

Logendran R, Carson S, Hanson E (2005). Group scheduling in flexible flow shops. International Journal of Production Economics, 96(2): 143–155

Logendran R, deSzoeke P, Barnard F (2006a). Sequence-dependent group scheduling problems in flexible flow shops. International Journal of Production Economics, 102(1): 66–86

Logendran R, Mai L, Talkington D (1995). Combined heuristics for bi-level group scheduling problems. International Journal of Production Economics, 38(2): 133–145

Logendran R, Nudtasomboon N (1991). Minimizing the makespan of a group scheduling problem: A new heuristic. International Journal of Production Economics, 22(3): 217–230

Logendran R, Salmasi N, Sriskandarajah C (2006b). Two-machine group scheduling problems in discrete parts manufacturing with sequence-dependent setups. Computers & Operations Research, 33(1): 158–180

Logendran R, Sriskandarajah C (1993). Two-machine group scheduling problem with blocking and anticipatory setups. European Journal of Operational Research, 69(3): 467–481

Low C, Lin W Y (2012). Single machine group scheduling with learning effects and past-sequence-dependent setup times. International Journal of Systems Science, 43(1): 1–8

Lu D, Logendran R (2013). Bi-criteria group scheduling with sequence-dependent setup time in a flow shop. Journal of the Operational Research Society, 64(4): 530–546

Lu Y Y, Wang J J, Wang J B (2014). Single machine group scheduling with decreasing time-dependent processing times subject to release dates. Applied Mathematics and Computation, 234: 286–292

Mahmoodi F, Dooley K J (1991). A comparison of exhaustive and non-exhaustive group scheduling heuristics in a manufacturing cell. International Journal of Production Research, 29(9): 1923–1939

Mahmoodi F, Dooley K J, Starr P J (1990). An investigation of dynamic group scheduling heuristics in a job shop manufacturing cell. International Journal of Production Research, 28(9): 1695–1711

Mendes N F M, Arroyo J E C, Villadiego H M M (2013). Local search heuristics for the flowshop sequence dependent group scheduling problem. In: 29th Latin American Computing Conference. Caracas: IEEE, 1–7

Neufeld J S, Gupta J N D, Buscher U (2015). Minimising makespan in flowshop group scheduling with sequence-dependent family set-up times using inserted idle times. International Journal of Production Research, 53(6): 1791–1806

Neufeld J S, Gupta J N D, Buscher U (2016). A comprehensive review of flowshop group scheduling literature. Computers & Operations Research, 70: 56–74

Nie L, Gao L, Hu Y D (2007). Prefix-gene-expression-programming-based algorithm for scheduling groups of jobs on a single machine. Computer Integrated Manufacturing Systems, 115(11): 2261–2268, 2275 (in Chinese)

Pan E, Wang G, Xi L, Chen L, Han X (2014). Single-machine group scheduling problem considering learning, forgetting effects and preventive maintenance. International Journal of Production

Research, 52(19): 5690–5704

Pan Q K, Gao L, Wang L (2022). An effective cooperative co-evolutionary algorithm for distributed flowshop group scheduling problems. IEEE Transactions on Cybernetics, 52(7): 5999–6012

Qin H, Han Y, Wang Y, Liu Y, Li J, Pan Q (2022a). Intelligent optimization under blocking constraints: A novel iterated greedy algorithm for the hybrid flow shop group scheduling problem. Knowledge-Based Systems, 258: 109962

Qin H, Zhang Z H, Bai D (2016). Permutation flowshop group scheduling with position-based learning effect. Computers & Industrial Engineering, 92: 1–15

Qin H X, Han Y Y, Liu Y P, Li J Q, Pan Q K, Han X (2022b). A collaborative iterative greedy algorithm for the scheduling of distributed heterogeneous hybrid flow shop with blocking constraints. Expert Systems with Applications, 201: 117256

Qin H X, Han Y Y, Zhang B, Meng L L, Liu Y P, Pan Q K, Gong D W (2022c). An improved iterated greedy algorithm for the energy-efficient blocking hybrid flow shop scheduling problem. Swarm and Evolutionary Computation, 69: 100992

Radharamanan R (1986). A heuristic algorithm for group scheduling. Computers & Industrial Engineering, 11(1): 204–208

Ren J, Ye C, Yang F (2021). Solving flow-shop scheduling problem with a reinforcement learning algorithm that generalizes the value function with neural network. Alexandria Engineering Journal, 60(3): 2787–2800

Rossit D A, Tohmé F, Frutos M (2018). The non-permutation flowshop scheduling problem: A literature review. Omega, 77: 143–153

Ruben R A, Mosier C T, Mahmoodi F (1993). A comprehensive analysis of group scheduling heuristics in a job shop cell. International Journal of Production Research, 31(6): 1343–1369

Salmasi N, Logendran R (2008). A heuristic approach for multi-stage sequence-dependent group scheduling problems. Journal of Industrial Engineering International, 4(7): 48–58

Salmasi N, Logendran R, Skandari M R (2010). Total flow time minimization in a flowshop sequence-dependent group scheduling problem. Computers & Operations Research, 37(1): 199–212

Salmasi N, Logendran R, Skandari M R (2011). Makespan minimization of a flowshop sequence-dependent group scheduling problem. International Journal of Advanced Manufacturing Technology, 56(5): 699–710

Schaller J (2001). A new lower bound for the flow shop group scheduling problem. Computers & Industrial Engineering, 41(2): 151–161

Schaller J E, Gupta J N D, Vakharia A J (2000). Scheduling a flowline manufacturing cell with sequence dependent family setup times. European Journal of Operational Research, 125(2): 324–339

Shahvari O, Salmasi N, Logendran R, Abbasi B (2012). An efficient tabu search algorithm for flexible flow shop sequence-dependent group scheduling problems. International Journal of Production Research, 50(15): 4237–4254

Shao Z, Shao W, Pi D (2021). Effective constructive heuristic and iterated greedy algorithm for distributed mixed blocking permutation flowshop scheduling problem. Knowledge-Based Systems, 221: 106959

Solimanpur M, Elmi A (2011). A tabu search approach for group scheduling in buffer-constrained flow shop cells. International Journal of Computer Integrated Manufacturing, 24(3): 257–268

Song H, Yi S, Wu C, Zhang S, Deng G, Liu P, Wei X (2020). Research on group scheduling of optimal setup uncorrelated parallel machine based on GATS hybrid algorithm. Journal of Chongqing University, 43(1): 53–63 (in Chinese)

Taghavi-fard M T, Javanshir H, Roueintan M A, Soleimany E (2011). Multi-objective group scheduling with learning effect in the cellular manufacturing system. International Journal of Industrial Engineering Computations, 2(3): 617–630

Tang H, Fang B, Liu R, Li Y, Guo S (2022a). A hybrid teaching and learning-based optimization algorithm for distributed sand casting job-shop scheduling problem. Applied Soft Computing, 120: 108694

Tang J, Haddad Y, Salonitis K (2022b). Reconfigurable manufacturing system scheduling: A deep reinforcement learning approach. Procedia CIRP, 107: 1198–1203

Tavakkoli-Moghaddam R, Javadian N, Khorrami A, Gholipour-Kanani Y (2010). Design of a scatter search method for a novel multi-criteria group scheduling problem in a cellular manufacturing system. Expert Systems with Applications, 37(3): 2661–2669

van der Zee D J (2013). Family based dispatching with batch availability. International Journal of Production Research, 51(12): 3643–3653

Villadiego H M M, Arroyo J E C, Jacob V V, dos Santos A G, Goncalves L B (2012). An efficient ILS heuristic for total flow time minimization in a flow shop sequence dependent group scheduling problem. In: 12th International Conference on Hybrid Intelligent Systems (HIS). Pune: IEEE, 259–264

Wang J B, Gao W J, Wang L Y, Wang D (2009). Single machine group scheduling with general linear deterioration to minimize the makespan. International Journal of Advanced Manufacturing Technology, 43(1–2): 146–150

Wang J B, Guo A X, Shan F, Jiang B, Wang L Y (2007). Single machine group scheduling under decreasing linear deterioration. Journal of Applied Mathematics & Computing, 24(1): 283–293

Wang J B, Wang J J (2014). Single machine group scheduling with time dependent processing times and ready times. Information Sciences, 275: 226–231

Wang J J, Liu Y J (2014). Single-machine bicriterion group scheduling with deteriorating setup times and job processing times. Applied Mathematics and Computation, 242: 309–314

Wang Y, Wang S, Li D, Shen C, Yang B (2021). An improved multi-objective whale optimization algorithm for the hybrid flow shop scheduling problem considering device dynamic reconfiguration processes. Expert Systems with Applications, 174: 114793

Wang Z Y, Pan Q K, Gao L, Wang Y L (2022). An effective two-stage iterated greedy algorithm to minimize total tardiness for the distributed flowshop group scheduling problem. Swarm and Evolutionary Computation, 74: 101143

Wilson A D, King R E, Hodgson T J (2004). Scheduling non-similar groups on a flow line: Multiple group setups. Robotics and Computer-integrated Manufacturing, 20(6): 505–515

Wu C, Wang L, Wang J (2021). A path relinking enhanced estimation of distribution algorithm for direct acyclic graph task scheduling problem. Knowledge-Based Systems, 228: 107255

Wu X, Cao Z (2022). An improved multi-objective evolutionary algorithm based on decomposition for solving re-entrant hybrid flow shop scheduling problem with batch processing machines. Computers

& Industrial Engineering, 169: 108236

Yan Y, Zhao C (2007). Single machine group scheduling with resource dependent setup times. Systems Engineering and Electronics, 333(6): 938–941 (in Chinese)

Yang D L, Chern M S (2000). Two-machine flowshop group scheduling problem. Computers & Operations Research, 27(10): 975–985

Yang D L, Kuo W H, Chern M S (2008a). Multi-family scheduling in a two-machine reentrant flow shop with setups. European Journal of Operational Research, 187(3): 1160–1170

Yang S J (2011). Group scheduling problems with simultaneous considerations of learning and deterioration effects on a single-machine. Applied Mathematical Modelling, 35(8): 4008–4016

Yang S J, Yang D L (2010). Note on “A note on single-machine group scheduling problems with position-based learning effect”. Applied Mathematical Modelling, 34(12): 4306–4308

Yang W H (2002). Group scheduling in a two-stage flowshop. Journal of the Operational Research Society, 53(12): 1367–1373

Yang W H, Liao C J (1996). Group scheduling on two cells with intercell movement. Computers & Operations Research, 23(10): 997–1006

Yang Y, Li X (2022). A knowledge-driven constructive heuristic algorithm for the distributed assembly blocking flow shop scheduling problem. Expert Systems with Applications, 202: 117269

Yang Y, Wang D Z, Wang D W, Wang H F (2008b). Single machine group scheduling problem with resource constraints and deteriorating jobs. Control and Decision, 23(12): 1413–1416, 1422 (in Chinese)

Yazdani Sabouni M T, Logendran R (2013a). A single machine carry-over sequence-dependent group scheduling in PCB manufacturing. Computers & Operations Research, 40(1): 236–247

Yazdani Sabouni M T, Logendran R (2013b). Carryover sequence-dependent group scheduling with the integration of internal and external setup times. European Journal of Operational Research, 224(1): 8–22

Yin N, Kang L, Wang X Y (2014). Single-machine group scheduling with processing times dependent on position, starting time and allotted resource. Applied Mathematical Modelling, 38(19–20): 4602–4613

Yoshida T, Nakamura N, Hitomi K (1977). A study of group scheduling. Journal of Japan Industrial Management Association, 28(3): 323–328

Yuan S, Li T, Wang B (2020a). A co-evolutionary genetic algorithm for the two-machine flow shop group scheduling problem with job-related blocking and transportation times. Expert Systems with Applications, 152: 113360

Yuan S, Li T, Wang B (2021). A discrete differential evolution algorithm for flow shop group scheduling problem with sequence-dependent setup and transportation times. Journal of Intelligent Manufacturing, 32(2): 427–439

Yuan S, Li T, Wang B (2022). Enhanced migrating birds optimization algorithm for hybrid flowshop group scheduling problem with unrelated parallel machines. Computer Integrated Manufacturing Systems, 28(12): 3912–3922 (in Chinese)

Yuan S, Li T, Wang B, Yu N (2020b). Model and algorithm for two-stage flow shop group scheduling problem with special blocking constraint. Control and Decision, 35(7): 1773–1779 (in Chinese)

Yue L, Guan Z, Saif U, Zhang F, Wang H (2016). Hybrid Pareto artificial bee colony algorithm for multi-objective single machine group scheduling problem with sequence-dependent setup times and learning effects. SpringerPlus, 5(1): 1593

Zandieh M, Dorri B, Khamseh A R (2009). Robust metaheuristics for group scheduling with sequence-dependent setup times in hybrid flexible flow shops. International Journal of Advanced Manufacturing Technology, 43(7–8): 767–778

Zandieh M, Hashemi A R (2015). Group scheduling in hybrid flexible flowshop with sequence-dependent setup times and random breakdowns via integrating genetic algorithm and simulation. International Journal of Industrial and Systems Engineering, 21(3): 377–394

Zandieh M, Karimi N (2011). An adaptive multi-population genetic algorithm to solve the multi-objective group scheduling problem in hybrid flexible flowshop with sequence-dependent setup times. Journal of Intelligent Manufacturing, 22(6): 979–989

Zhang B, Pan Q, Meng L, Lu C, Mou J, Li J (2022a). An automatic multi-objective evolutionary algorithm for the hybrid flowshop scheduling problem with consistent sublots. Knowledge-Based Systems, 238: 107819

Zhang C, Xu W, Liu J, Liu Z, Zhou Z, Pham D T (2019). A reconfigurable modeling approach for digital twin-based manufacturing system. Procedia CIRP, 83: 118–125

Zhang X, Liao L, Zhang W, Cheng T C E, Tan Y, Ji M (2018). Single-machine group scheduling with new models of position-dependent processing times. Computers & Industrial Engineering, 117: 1–5

Zhang Z Q, Qian B, Hu R, Jin H P, Wang L, Yang J B (2022b). A matrix-cube-based estimation of distribution algorithm for blocking flow-shop scheduling problem with sequence-dependent setup times. Expert Systems with Applications, 205: 117602

Zhao F, Shao D, Wang L, Xu T, Zhu N, Jonrinaldi (2022). An effective water wave optimization algorithm with problem-specific knowledge for the distributed assembly blocking flow-shop scheduling problem. Knowledge-Based Systems, 243: 108471

Zhao F, Zhang L, Cao J, Tang J (2021). A cooperative water wave optimization algorithm with reinforcement learning for the distributed assembly no-idle flowshop scheduling problem. Computers & Industrial Engineering, 153: 107082

Zheng Y, Xie S, Qian W (2014). Hybrid differential evolution algorithm for FSDGS problem with limited buffers. Computer Integrated Manufacturing Systems, 20(8): 1941–1947 (in Chinese)

Zhou Y, Miao J, Yan B, Zhang Z (2021). Stochastic resource-constrained project scheduling problem with time varying weather conditions and an improved estimation of distribution algorithm. Computers & Industrial Engineering, 157: 107322

Zhu Z, Sun L, Chu F, Liu M (2011). Single-machine group scheduling with resource allocation and learning effect. Computers & Industrial Engineering, 60(1): 148–157

Zolfaghari S, Liang M (1999). Jointly solving the group scheduling and machining speed selection problems: A hybrid tabu search and simulated annealing approach. International Journal of Production Research, 37(10): 2377–2397