

Ying, Kuo-Ching; Pourhejazy, Pourya; Lin, Zhi-Rong

Article

Scheduling with sequence-dependent setup times in short-term production planning: A main path analysis-based review

Operations Research Perspectives

Provided in Cooperation with:

Elsevier

Suggested Citation: Ying, Kuo-Ching; Pourhejazy, Pourya; Lin, Zhi-Rong (2025) : Scheduling with sequence-dependent setup times in short-term production planning: A main path analysis-based review, Operations Research Perspectives, ISSN 2214-7160, Elsevier, Amsterdam, Vol. 14, pp. 1-17, <https://doi.org/10.1016/j.orp.2025.100340>

This Version is available at:

<https://hdl.handle.net/10419/325817>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<https://creativecommons.org/licenses/by/4.0/>



Scheduling with sequence-dependent setup times in short-term production planning: A main path analysis-based review

Kuo-Ching Ying^a, Pourya Pourhejazy^{b,*}, Zhi-Rong Lin^{a,c}

^a Department of Industrial Engineering and Management, National Taipei University of Technology, Taipei 10608, Taiwan

^b Department of Industrial Engineering, UiT- The Arctic University of Norway, Lodve Langesgate 2, Narvik 8514, Norway

^c CSBC Corporation, Zhonggang Rd., Xiaogang Dist., Kaohsiung City 81234, Taiwan

ARTICLE INFO

Keywords:

Scheduling
Setup time
Main path analysis
Cluster analysis
Systematic review

ABSTRACT

The role of setup times in production planning and control was recognised in the late 1960s. Since then, a growing number of scheduling problems have accounted for sequence-dependent setup time variables. This study aims to provide a systematic review of setup times in the short-term production planning literature, using an objective, algorithm-based approach. The Main Path Analysis (MPA) and Cluster Analysis (CA) methods are employed to identify patterns of knowledge development and the most significant advancements in the field. Over 2100 articles published between 1986 and 2024 were considered in the review. The seminal articles contributing to the advances in setup times for production scheduling are reviewed. Meanwhile, the core optimisation technologies, model characteristics, and emerging issues at different stages of literature development are discussed. The key extensions of the main path are further explored to identify secondary research interests in the field. Twenty-two research themes were identified to provide an overall perspective and shed light on the technical features and challenges. Finally, future research directions are suggested based on the outcomes of this systematic review.

1. Introduction

Non-value-adding movements or delays on the shop floor are sources of waste in production, which can be minimised through well-informed production planning decisions; setup scheduling is one prime example [1]. The supply chain implications of ignoring setup times in production planning include increased inventory pressure, delays in delivery times, and, in the worst cases, production bottlenecks, supply shortages, and bullwhip effects [2]. In some application areas, setup times are more notable than in others, making it necessary to consider them in production scheduling. In general, effective setup planning helps reduce production downtime and operational costs, improve the flexibility of production batches, increase production capacity, and ultimately enhance production efficiency.

Setups are prevalent where resources must be used for diverse purposes. Scheduling with setup considerations is widely applied in both service and manufacturing environments and is also employed in modern areas such as computer systems and synchronous circuits. In a manufacturing environment, a setup refers to the operations required to switch from one job or product to another. This may include preparing

tools and materials, cleaning workstations and machinery, replacing equipment, adjusting machines, and more. The incorporation of setup time scheduling into cleaner production practices has been shown to not only reduce energy consumption [3] but also to minimise waste [4], thereby enhancing overall production efficiency. This has also been emphasized in just-in-time production, optimised production technology, group technology, and cellular manufacturing [5].

As a technical term in the scheduling literature, the Sequence-Dependent Setup Time (SDST) refers to the time required for implementing preparatory operations. In this definition, the setup time for a certain job or task depends not only on the characteristics of the task itself but also on the tasks performed before it. The optimisation of SDSTs consists of (1) planning the job sequence such that if two adjacent jobs can share certain preparatory steps, queuing them together may reduce the total setup time; (2) prioritising the jobs in a way that if an urgent task has to be carried out immediately, the preparation process for other tasks might have to be postponed; (3) utilizing resources such as manpower, equipment, and tools effectively to ensure that setup operations can be completed on time; and (4) employing coping strategies to deal with the impact of uncertainty and changes, such as

* Corresponding author.

E-mail address: pourya.pourhejazy@uit.no (P. Pourhejazy).

<https://doi.org/10.1016/j.orp.2025.100340>

Received 23 September 2024; Received in revised form 8 April 2025; Accepted 8 April 2025

Available online 9 April 2025

2214-7160/© 2025 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

machine failures, employee absences, and sudden demand surges affecting the actual setup time.

Since its recognition in the 1960s, considering SDSTs in production planning has remained an evolving research topic. Setup time was initially introduced and tested within single-machine production environments and as a static parameter [6]. SDSTs have most recently been incorporated into advanced scheduling problems, such as the energy-aware distributed hybrid Flow Shop Scheduling Problem (FSSP) [7], parallel batch processor lot-sizing and scheduling [8], reentrant hybrid FSSP [9], dual resource-constrained flexible Job Shop Scheduling Problem (JSSP) [10], distributed two-stage assembly FSSP [11], distributed permutation FSSP [12], and distributed heterogeneous hybrid blocking FSSP with flexible assembly [13]. Recent studies have also integrated SDST considerations with energy-efficiency optimisation, demonstrating that adjusting sequences based on setup requirements can lead to improved energy performance in diverse production scenarios [7].

In the first comprehensive reviews of the scheduling literature, Ref. [6] showed that more research exists on single-machine scheduling problems with setup time than in other production settings. Such a comprehensive review of the literature provides an in-depth understanding of the developments in the field and requires a substantial investment of time. Ref. [5] covered scheduling studies on random setup time in a wider set of factory settings. The authors classified these studies based on scheduling (1) with and without batching considerations; (2) with sequence-independent setup times and SDSTs; and (3) shop floor settings, including single-machine, parallel machines, flow shop, no-wait flow shop, flexible flow shop, job shop, and open shop. The third comprehensive review [14] covered static, dynamic, deterministic, and stochastic scheduling problems and classified them according to factory settings and operational considerations for family and non-family products. The three comprehensive reviews covered studies from 1960 to 2014, with manual reviews of 200, 300, and 500 articles, respectively. To our knowledge, the most recent comprehensive review dates back to 2016, when Ref. [15] surveyed the literature on JSSP that included setup considerations. This review categorized the literature into JSSPs with non-batch (job) setup times and JSSPs with batch setup times while also providing an in-depth analysis of the exact, hybrid, and heuristic solution methods used under each category.

The existing comprehensive reviews on scheduling with setup time have relied on traditional review methods, which involve manual data collection and analysis, subjecting them to the authors' individual judgement. Moreover, a considerable amount of time has passed since the most recent broad review on scheduling with setup time was last conducted. It is, therefore, appropriate to update our understanding of setup times in the scheduling literature by employing advanced data analysis approaches. This study employs an objective, algorithm-based approach to broadly review SDSTs in the scheduling literature, offering a more systematic and data-driven perspective compared to traditional reviews. Through Main Path Analysis (MPA) and Cluster Analysis (CA), this study explores the historical trends, themes, and dynamics of SDSTs in the scheduling literature, overcoming limitations in previous reviews by eliminating the biases associated with manual data collection. This approach also differs from past methods by offering a more precise and systematic analysis of the literature. For this purpose, MainPath 465 software was utilized for data processing and analysis. Subsequently, Pajek software was employed for further analysis of the citation network and the identification of the Knowledge Dissemination Trajectory (KDT) and Knowledge Development Clusters (KDC). Lastly, VOSviewer was used to visualize keywords within different literature clusters and identify development patterns in the literature. This systematic review approach highlights the core optimisation technologies, model characteristics, and emergent issues at various stages of literature development [16].

The remainder of this review article is organized in Sections 2–6. Section 2 explains the materials and methods used in this study. Sections

3 and 4 present the global MPA of the development trajectory and its branches, respectively. Section 5 analyses the research clusters to identify significant research themes based on keywords and reviews the most recent developments within each thematic. Finally, Section 6 summarizes the key findings and offers insights into future research directions.

2. Materials and methods

2.1. Data collection and processing

The Web of Science databases were used for data collection. The platform provided resources for exploring and analyzing scientific literature, including academic journals, conference papers, patents, and book chapters published between 1986 and the end of 2023. The following search protocol (((TS=(setup*) OR TS=(set-up*) OR TS=(removal*)) AND TS=(scheduling*)) AND (TS=(sequence))) yielded a total of 2195 items. From this set, 41 retrospective documents, early access works, and isolated documents were excluded. Review articles typically have a higher number of citations and, therefore, can introduce a subjective impact on the analysis of the KDTs. Early access documents were excluded because they could result in loops. Finally, 168 documents that neither cited other documents nor were cited by others were removed from the dataset. This data collection and processing procedure is illustrated in Fig. 1.

After finalizing the dataset, the option "Complete records and cited references" is used to export the required data. The data processing stage begins with importing the compiled documents into the software to calculate the traversal weight of all links in the citation network. First, the global and key-route MPA are considered to review the seminal KDTs on SDSTs in the scheduling literature and to identify the development of the most influential research trends and directions. Next, the cluster analysis method is used to divide the network into several clusters, each of which represents a sub-field of scheduling with SDSTs, from which additional insights can be derived. Considering the identified sub-fields, VOSviewer software is ultimately used to complete the keyword analysis and to visualize the clusters.

Precision and the Digital Object Identifier Percentage (DOI Percentage) indicators are considered to evaluate whether the dataset is representative. Precision determines whether the number of nodes constitutes a high proportion of the total number of articles in the original database. The DOI Percentage measures the dataset quality as the ratio of the number of cited documents in the collected dataset to the total number of citations in the original database. These values reflect the representativeness of the dataset.

Considering a network with 1973 nodes (sources: 198, sinks: 478, isolates: 0, intermediates: 1297), Precision is calculated as Network Size divided by the Number of Articles, yielding $1973 / 2141 = 0.92$. This result confirms that the database retains a strong correlation after screening. To calculate the DOI Percentage, the text files of 2141 documents were imported into the MPA software for analysis of their citation relationships. The results showed that the total citation records amounted to 61,499, while 74,856 citation relationships were identified in the dataset. The number of citations accounted for a DOI Percentage of 0.82 (DOI total = 61,499, CR total = 74,856) relative to the total number of citations. According to Ref. [17], Precision and DOI Percentage values exceeding 70 % confirm that the database used for analysis is representative.

2.2. Main path and cluster analysis

This study employs MPA, a citation-based method, to conduct a systematic and unbiased review of setup times within the short-term production planning literature. MPA begins by constructing a citation network in which articles are represented as nodes and citation relationships are depicted as links. As can be seen in Fig. 2, this creates an

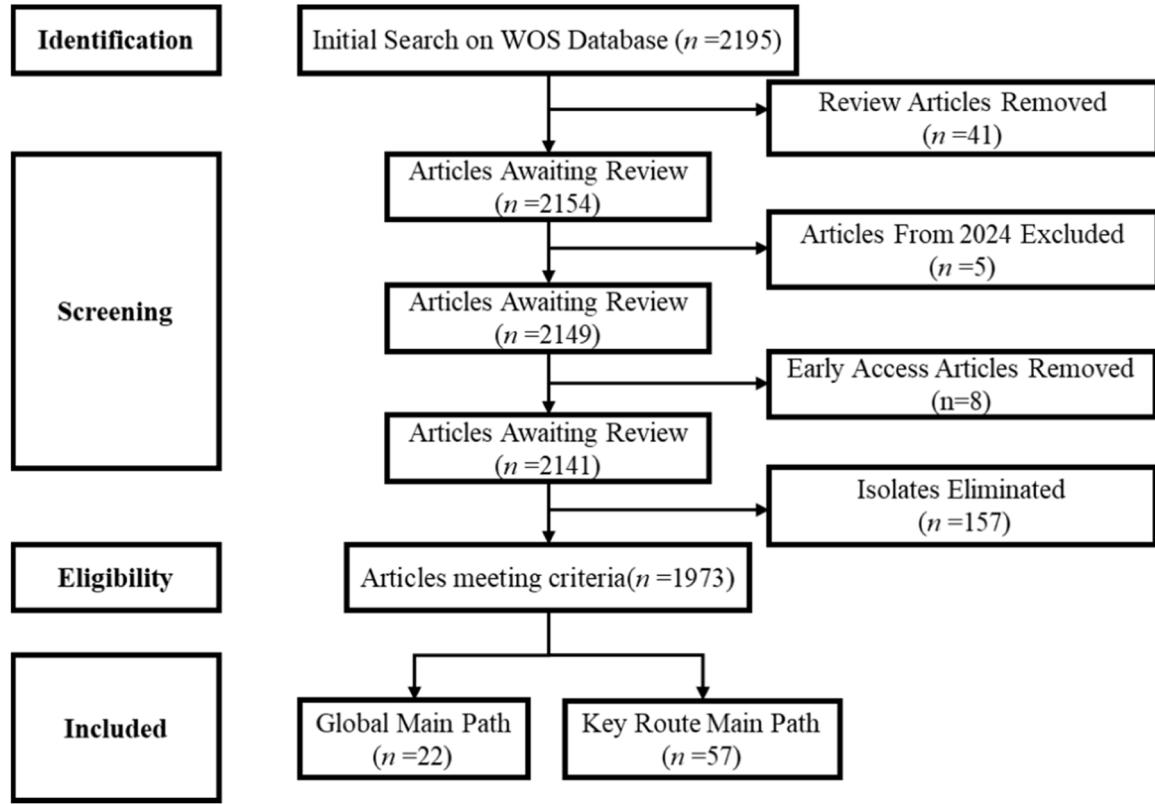


Fig. 1. Data collection and processing procedure.

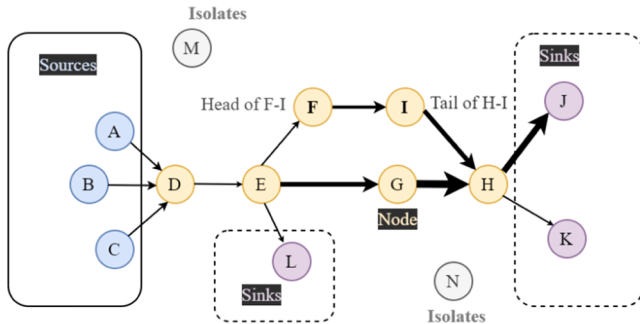


Fig. 2. Example of a citation network.

acyclic directed network with sources, sinks, intermediates, isolates, and citation chains. Source nodes represent the origin of the citation network, sink nodes indicate its endpoints, and intermediate nodes constitute the paths connecting the origin to the endpoints.

In this context, a path may consist of multiple links (i.e., citations), with arrows directed from the cited document to the citing document. The MPA method is first employed to identify all possible path(s) from source point(s) to sink point(s) in two steps: establishing a weighted network and identifying the ‘main paths’. Three methods can be used to establish a weighted network characterised by the traversal weight of each link: (1) Search Path Count (SPC), which calculates the number of times a link is crossed when all possible paths from all sources to all sinks are considered. (2) Search Path Link Count (SPLC), which calculates the number of crossings when all possible paths from the ancestors of a tail node to all sinks are considered. (3) Search Path Node Pair (SPNP), which sums the number of crossings from all possible paths originating from the ancestors of a tail node to the descendants of a head node. The SPLC scheme is considered the most effective method for representing the KDT in the literature [17] and is, therefore, used in this study to

determine the traversal weight of each link in the citation network.

In the weighted citation network, the most traversed paths are identified as the network’s ‘main path’ and are selected for in-depth analysis. Several methods can be used to identify the main paths in the weighted citation network. We employed the global and key-route MPA methods, which are the most commonly used in the literature. The former identifies the path with the greatest total weight of all links, representing the most important KDT in the advances of setup times across the scheduling literature. In contrast, the latter considers the most cited link(s) as the basis for exploring all paths to the source and sink nodes. The key-route MPA method facilitates the exploration of inter-related development trajectories, ensuring the inclusion of all important articles in the in-depth analysis. This methodology, while innovative and rigorous, is complex. Interested readers can consult foundational works on MPA [17], where detailed explanations of MPA’s computational mechanics and applications are provided.

Finally, CA is employed in this study to conduct a thematic analysis of the published articles on scheduling problems with SDSTs. CA, developed by [18], is a tree-based categorization approach that uses the number of shortest paths between all network nodes to identify KDCs. CA calculates the so-called Edge Credit = $(1 + \sum \text{Incoming Edge Credit}) \times \frac{\text{Score of Destination}}{\text{Score of Start}}$ for each node, starting from the source to the sink nodes in the citation network. The edge credit score of a particular node represents the number of steps from the source of KDT to the current node in the citation network. CA removes the edge(s) with the highest total score to form clusters.

2.3. Parameter selection

For both the MPA and CA methods, the selection of algorithm parameters and software settings was guided by theoretical insights from the literature and empirical sensitivity tests conducted on our dataset. In implementing MPA, we evaluated several traversal weight calculation methods, namely SPC, SPLC, and SPNP. After comparing these methods,

we selected the SPLC method because it has been identified as the most effective approach for representing KDT in the literature [17]. In addition, this study tested key-route values of 10, 15, 20, 25, 30, and 35 to determine the most appropriate key-route main path representation. The test results were as follows: a key-route value of 10 produced 30 articles, 15 produced 47 articles, 20 produced 49 articles, 25 produced 51 articles, 30 produced 57 articles, and 35 produced 58 articles. Based on these results, which show a trend of diminishing marginal returns, the study concludes that the most popular articles in this field are concentrated on the key-route main path, and further testing beyond these values would provide minimal additional insights.

Therefore, considering the extent of diminishing returns, a key-route value of 30 is deemed most appropriate. For the CA, this study imported 1937 literature records into the MainPath480 subroutine GroupFinder and performed clustering using the Edge-Betweenness Clustering algorithm. Finally, the study applied a key-route search with a key-route value of 5 to the clustering results to identify the key-route main paths, providing insights into the focal topics of each cluster. These parameter choices and software settings ensure that the resulting analysis is robust and reproducible.

3. Results of the main path analysis

After excluding the isolated points from the formal analysis, a total of 22 articles were included to form the global main path (shown in Fig. 3). In this figure, the size of the arrow corresponds to the SPLC value associated with the corresponding link. Each node represents an article and is labeled with the first author's last name, followed by the initials of the other authors' last names (if applicable) and the year of publication. The main path articles are discussed in four distinct developmental phases:

- (1) 1992–2008: FSSP with SDSTs;
- (2) 2009–2019: Hybrid/Flexible FSSP with SDSTs;
- (3) 2020–2022: Distributed permutation FSSP with SDSTs; and
- (4) 2022–2023: Flow shop Group Scheduling Problem (GSP) with SDSTs.

Most studies on the main path considered the maximum completion time (makespan) as the optimisation criterion, but some studies focus on minimizing the total weighted earliness and tardiness. These studies introduced practical constraints, such as learning effects, transportation time, and due dates, among others. We provide a detailed discussion of these articles in the following subsections.

3.1. 1992 to 2008

Simons1992 [19] initiated the main path of FSSPs with SDSTs by exploring how arranging jobs while considering the possible setups could minimise the makespan. He developed four heuristic algorithms, among which Total and Setup, both based on the Travelling Salesman Problem (TSP), outperformed the minimum idle time and minimum completion time rules in terms of average performance, worst-case scenarios, standard deviation, and the frequency of best solutions.

Ríos-MercadoB1998 [20] continued Simons's research by introducing the NEHT-RB and Greedy Randomized Adaptive Search Procedure (GRASP) heuristics and comparing them with Simons's SETUP for solving FSSPs with SDSTs. Minimising the makespan, they showed that NEHT-RB and GRASP outperformed SETUP when the setup time was shorter than the processing time. They also showed that GRASP was slower than SETUP and NEHT-RB but offered more diverse solutions. Ríos-MercadoB1990 [21] proposed an improved TSP-based heuristic with a hybrid cost function to balance the impact of setup time and fitness value in FSSPs with SDSTs. They found that the new algorithm could produce better solutions in most cases and was more efficient than GRASP. The same authors, Ríos-MercadoB1990 [22] proposed a new

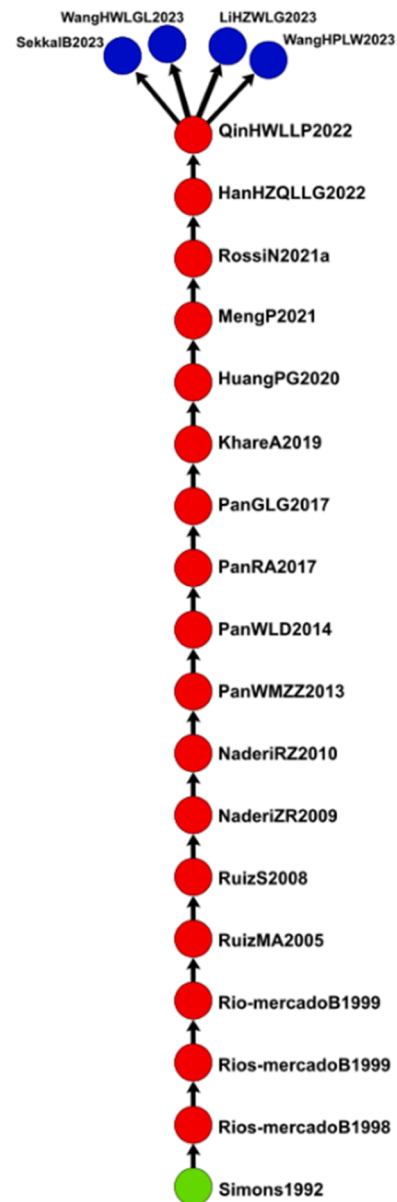


Fig. 3. The global main path in the literature on scheduling with setup time.

branch-and-bound method for solving permutation FSSPs with SDSTs. Their approach introduced lower bound calculation methods, including the generalized lower bound and machine-based lower bound, for Non-Linear Programming (NLP). They used a new criterion to avoid unnecessary branches and to select the subproblem with the smallest lower bound for branching. The experimental results showed that their algorithm was superior to the Linear Programming (LP) methods concerning efficiency and effectiveness.

RuizMA2005 [23] sought to minimise the makespan in FSSPs with SDSTs using two advanced Genetic Algorithms (GAs), one of which was hybridized with a local search method. They compared these with improved versions of Simulated Annealing (SA), Tabu Search (TS), Variable Neighborhood Search (VNS), and Iterated Local Search (ILS). RuizS2008 [24] addressed FSSPs with SDSTs, which are commonly encountered in wafer manufacturing. They hybridized the IG with a local search module (IG_RSLs) to minimise the makespan and total weighted tardiness. Comparing the basic and improved algorithms with 14 other solution methods showed that IG was superior to other methods in terms of solution quality and efficiency and led to the discovery of

new best-known solutions.

3.2. 2009 to 2019

NaderiZR2009 [25] developed a hybrid SA with a local search module to solve hybrid FSSPs with SDSTs, aiming to minimise the makespan and maximum tardiness. They introduced several improvements to the SA algorithm, such as the migration mechanism and the giant leap. The algorithm performed exceptionally well, regardless of the instance type. NaderiZR2009 [26] introduced the hybrid flexible FSSP with SDSTs. They developed the modified dynamic dispatching rule and the ILS algorithm to minimise the makespan. MDDR allocates jobs to machines based on their earliest completion times while avoiding longer setup times. ILS applies local search and perturbation operators to improve the exploration power of the algorithm. The authors compared these algorithms with the state-of-the-art to demonstrate their effectiveness.

PanWMZZ2013 [27] developed the Artificial Bee Colony (ABC) algorithm to solve hybrid FSSPs with SDSTs, which are prevalent in the steelmaking industry, minimising waiting time and the casting start time with early/tardy penalties. The problem involves three consecutive stages with parallel machines. They introduced a heuristic method to generate the initial solution and adopted new neighbourhood search mechanisms. PanWLD2014 [28] improved the ABC algorithm through hybrid representation, enhanced search strategies, and tested a discrete variant to solve the hybrid FSSP with SDSTs, considering the makespan. They utilised 24 heuristic rules to generate initial solutions, tested a new control parameter to balance exploration and exploitation, and introduced an enhanced search strategy to prevent the algorithm from falling into local optima. PanRA2017 [29] explored hybrid FSSP with SDST and time-window constraints and developed the ILS and IG algorithms to minimise weighted earliness and tardiness costs. They introduced a novel solution representation method that only sorts the jobs in the first stage and uses a dispatch rule to determine the allocations and the job orders in later stages. PanGLG2017 [30] studied hybrid FSSP with SDST and developed nine heuristics and metaheuristics to solve the problem, considering the makespan. These algorithms are variants of ILS, IG, Improved Fruit Fly Optimisation (IFFO), Improved Migrating Birds Optimisation (IMBO), and the discrete ABC. They found that the discrete ABC outperformed the benchmarks.

KhareA2019 [31] also solved hybrid FSSP with SDST and time-window constraints by developing three population-based optimisation algorithms, considering the total weighted cost of the early and tardy jobs. The Hybrid Squirrel Search Algorithm (HSSA), Opposition-Based Whale Optimisation Algorithm (OBWOA), and Discrete Gray Wolf Optimisation (DGWO) were coupled with VNS, Hybrid Local Search (HLS), and Opposition-Based Learning (OBL) methods to achieve the study objectives.

3.3. 2020 to 2022

HuangPG2020 [32] studied the distributed permutation FSSP with SDSTs to minimise the makespan using an improved version of the IG algorithm. The problem involved job assignments to a set of parallel plants, each of which consists of multiple machines and operates as a process plant. The improved IG employed a restart scheme and a control parameter to regulate solution diversity and help the algorithm escape from local optima.

MengP2021 [33] introduced lot streaming and carryover SDST features in distributed permutation FSSPs with heterogeneous factories. They proposed a Mixed-Integer Linear Programming (MILP) model and the Enhanced ABC algorithm to solve the problem with the aim of minimising the makespan in all factories. The heterogeneous variant of distributed permutation FSSP considers multiple different factories with unique process types and multiple machines. Carryover SDSTs means that the feeder on the machine needs to be replaced or adjusted to

prepare the required parts before processing the first sub-batch of a new job sequence. This article was the first to study the decentralized heterogeneous scheduling problem that combines batch partitioning and cumulative SDST, demonstrating high practicality.

RossiN2021a [34] studied the mixed-no-idle version of distributed FSSP with SDSTs. They proposed a mathematical formulation along with a constructive heuristic to solve it. The developed solution algorithm outperformed the state-of-the-art on a large new dataset developed by the authors RossiN2021a [34]. HanHZQLLG2022 [35] investigated the distributed blocking FSSP with SDSTs, aiming to minimise energy consumption costs while balancing resources across factories using an improved version of the IG algorithm. Their IG algorithm was improved through integration with VNS. The algorithm includes an efficient initialization heuristic, two distinct local search strategies, and a learning-based VNS strategy. The study introduced a scheduling model of high practical significance.

QinHWLLP2022 [36] studied the blocking hybrid flow shop GSP, characterised by a factory setting without a temporary storage area. They proposed a novel IG algorithm to minimise the makespan. This group of studies demonstrates that scholars' focus in solving different variants of the Distributed Scheduling Problem (DSP) has gradually shifted toward developing adaptable algorithms and away from basic optimisation problems to more complex, real-world production settings.

3.4. 2022 to present

The study by QinHWLLP2022 [36] on blocking hybrid flow shop GSP inspired four new research directions. SekkalB2023 [37] studied the flow shop GSP with SDSTs, incorporating learning effects and transportation time. They aimed to simultaneously minimise both the makespan and energy consumption using an MILP model and the Multi-Objective Simulated Annealing (MOSA) algorithm. They considered the practical case of forged connecting rods to demonstrate the impact of grouping technology and learning effects on production scheduling.

WangHWLGL2023 [38] studied the distributed flow shop GSP for minimising the makespan. They proposed an MILP model along with a two-stage Iterative Greedy (tIG) algorithm to solve it. They suggested that the distributed flow shop GSP can be optimised by solving three specific sub-problems. The tIG algorithm benefits from two collaborative neighbourhood search strategies across and within factories, as well as two enhanced search strategies across and within groups. Using a dataset of 810 test instances, they demonstrated that tIG algorithm outperformed the state-of-the-art algorithms.

LiHZWLGL2023 [39] studied a hybrid FSSP with SDSTs, incorporating batch processing machines, variable sub-batches, and transportation time, with the objective of maximising the total revenue. The paper proposed an MILP formulation and developed a new collaborative IG algorithm featuring a novel destruction-construction strategy to control sub-batches during the batch processing stage. They also introduced a dynamic acceptance criterion to balance the algorithm's exploitation and exploration capabilities.

Finally, WangHPLW2023 [40] studied hybrid flow shop GSP and proposed several MILP models as well as a constraint programming approach to minimise the makespan while addressing various practical constraints. Their experiments showed that two of the formulations perform significantly better, highlighting the effectiveness of combining sequential GSP with positional adjacency modelling of jobs within the groups. Overall, the most recent studies have focused on group scheduling, addressing the complexity and diversity of GSP.

4. Results of the key extension analysis

After identifying the main KDTs in the SDST literature, investigating the key extensions from this path will offer an overview of the innovative trends. This section elaborates on the main development branches,

considering a total of 57 documents from Keyroute30 (See Section 2.2); the overview is exhibited in Fig. 4.

The key branches are formed along three developmental stages (represented by solid blue lines), taking into account the time factor of the separation/integration nodes. Dotted frames are used to delineate detailed branching paths within each stage. Node colours signify the problem of interest, with red, green, and blue representing studies on batch scheduling, flow shop, and GSPs, respectively. The light colours identify the subcategories. The pink colour refers to the Parallel Machine Scheduling Problem (PMSP), and light green, bright green, and dark green correspond to permutation, hybrid, and distributed FSSPs, respectively. Finally, the node's colour at the intersection point is determined based on the document content.

Since the articles from the global main paths have already been reviewed, this section focuses on the remaining 35 documents that constitute the branches in Fig. 4. The figure shows three new source nodes, i.e., SinghF1987 [41], OvacikU1993 [42], and Sivrikaya-serifogluU1999 [43], in addition to those on the global main path. The three new source nodes are from the red categories; the resulting branch lines from the source nodes merge into other paths during the development process but eventually converge into the global main path. The dark and light green paths branch out from the middle and sides of the global main path and later converge back into it.

Overall, the key extensions initially revolved around FSSP and eventually developed into flow shop GSPs; this highlights that the most influential production planning studies on SDSTs focus primarily on the flow shop setting. The development branches in the first stage explore three themes: (1) The light green path, developed from the source point of Simons1992 [19] centres around FSSP. (2) The pink-to-green shift in the left-hand-side branch, which starts from OvacikU1993 [42] and Sivrikaya-serifogluU1999 [43], depicts the transformation of PMSPs into FSSP. (3) The red branch line on the right-hand side, initiated by SinghF1987 [41], branches out into two sub-branches: the former develops from FrançaGLM1996 [44] and depicts a transformation from PMSPs to FSSPs, while the latter represents a shift from the single-machine batch scheduling problem to the parallel-machine batch scheduling problems. We now delve deeper into the development branches.

4.1. Branch I

The first source node, Ovacik1993 [42], solved the PMSP with SDSTs to minimise the makespan and the maximum delay time. They assumed that the setup time would not exceed the processing time of the work order, inspired by the final test stage in semiconductor manufacturing, where different batches of circuits need to be tested at various

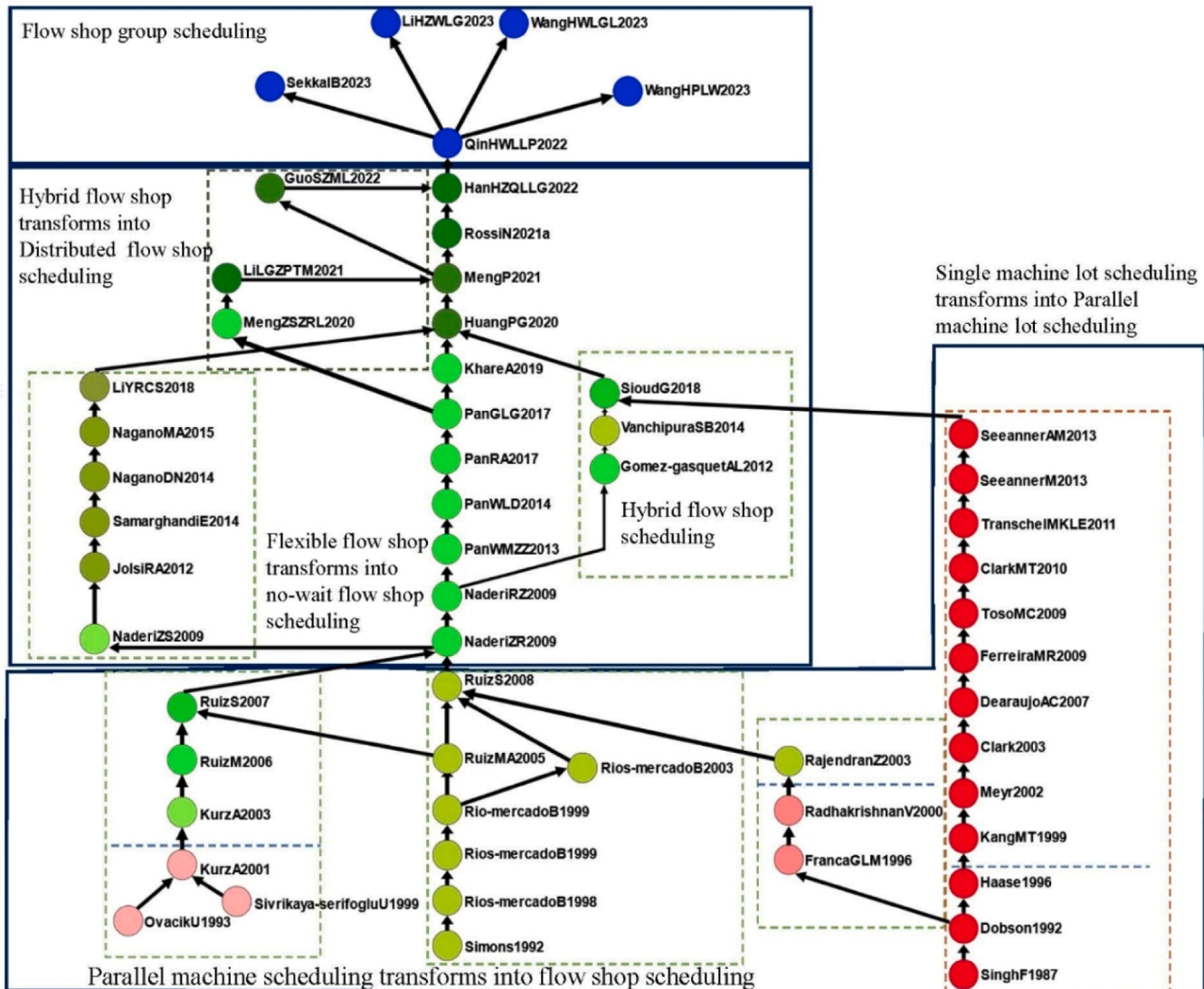


Fig. 4. The key branches emerged from the main path in the SDSTs literature.

temperatures (i.e., temperature-dependent changes in SDSTs). They derived the worst-case error bounds of arbitrary and specific-list scheduling algorithms for scheduling with bounded SDSTs and proved that the error bounds of their proposed algorithms were accurate.

The other source node, Sivrikaya-serifogluU1999 [43], also studied the PMSP with SDSTs but considered total earliness and delay costs as the optimisation objective. They developed a GA with novel crossover operators to solve the problem. By comparing it with other GAs, they found that their developed method outperformed the partially mapped crossover.

Combining the directions of these two source points, Kurza2001 [45] explored PMSP with SDST and release time, aiming to minimise the makespan. They developed and tested four heuristic algorithms to approximate the optimal solution. They also analysed the factors impacting the problem complexity and tested new techniques for obtaining lower bounds. Kurza2003 [46] began to investigate how to arrange a set of jobs to minimise the makespan in the flexible FSSP with SDSTs. They proposed three heuristics: the Cyclic Heuristic, the Multiple Insertion Heuristic, and Johnson's Rule-based Heuristic. Their experimental analysis showed that the algorithm based on Johnson's rule performed best in terms of average loss, standard deviation, and maximum loss, while the insertion algorithm achieved better solutions based on the frequency of finding the minimum loss solution. RuizM2006 [47] proposed a novel optimisation method based on GA to solve hybrid FSSPs involving unrelated parallel machines, SDSTs, and machine adaptation. They conducted an extensive numerical analysis using data from the ceramic tile manufacturing industry, demonstrating that their algorithm outperformed the state-of-the-art.

Finally, RuizS2007 [24] investigated permutation FSSP with SDSTs, which are prevalent in wafer manufacturing. They extended the IG algorithm to maximise the number of processed jobs, minimise the makespan, and minimise the makespan of non-bottleneck machines. They introduced a local search mechanism and an SA-inspired acceptance criterion to enhance the performance of the IG algorithm. The developed algorithm outperformed 14 solution methods and identified new best solutions for some test instances.

Overall, this research branch signifies the expansion of the research scope on SDSTs from PMSPs to more complex FSSPs, as well as a gradual advancement of solution algorithms, from list scheduling algorithms to the IG algorithm and heuristics.

4.2. Branch II

This branch illustrates the convergence of two paths in RuizS2008 [24]. One path represents a brief development from Ríos-MercadoB1999 [22] to Ríos-MercadoB2003 [48], while the other path extends from SinghF1987 [41] to Dobson1992 [49], which further branches into FrançaGLM1996 [44]. The KDT between Simons1992 [19] and RuizS2008 [24] has already been explained in Section 4.1. This subsection focuses on the paths from Ríos-MercadoB1999 [22] and Dobson1992 [49] in two distinct parts.

The first part (i.e., Simons1992 [19]→Ríos-MercadoB2003 [48]) is formed around FSSPs. The second part of this branch (i.e., FrançaGLM1996 [44]→RajendranZ2003 [50]) begins with PMSPs and then transitions to FSSP. In the first part, Ríos-MercadoB2003 [48] studied FSSP-SDST to minimise the makespan. They proposed two MILP models based on the polyhedral structure of the asymmetric TSP and the Linear Ordering Problem (LOP) and proved that the features and inequalities of the asymmetric TSP and the LOP can be directly applied to solving the scheduling issues. Numerical experiments showed that their proposed method could improve the lower bound in LP and solve medium-sized instances within a reasonable time.

In the second part, FrançaGLM1996 [44] extended the problem developed by Dobson1992 [49] to the Multiprocessor Scheduling Problem (MSP), which refers to identical PMSP-SDST for the makespan minimisation. They developed a TS algorithm that could adapt to

asymmetric setup times and, hence, improve the solution quality. The developed solution method outperformed the Nearest-Neighbor Heuristic algorithm.

RadhakrishnanV2000 [51] studied PMSP-SDST while minimising earliness and tardiness costs and developed an SA algorithm to find near-optimal solutions to the problem. Through a series of computational experiments, the performance of the SA method was evaluated and compared with that of a local search-based heuristic. RajendranZ2003 [50] discussed the FSSP-SDST for minimising the total weighted flowtime and delays and introduced a new heuristic algorithm for approximation. In their model, factors such as inventory or holding costs, contracting costs, and customer satisfaction were considered.

4.3. Branch III

The source node, SinghF1987 [41], studied the SMSP-SDST considering multiple products and proposed a three-stage optimisation framework. Their method identified a feasible production schedule with a minimum sum of inventory and maintenance costs within a given planning period. Dobson1992 [49]'s Cyclic Lot Scheduling Problem (CLSP) was a continuation of this development path.

Dobson1992 [49] studied CLSP with SDSTs. In this problem, the production batch size and production sequence are fixed, and the lead time varies depending on the characteristics of the products. They proposed a mixed-integer programming model and a local-search-based heuristic robust to different SDST configurations. The study of Haase1996 [52] is about capacitated batch scheduling; they proposed a new mathematical formulation called the capacity-limited and sequence-dependent batch production problem. Their method differs from earlier models by allowing continuous batch sizes, multiple setup times, and idle times. They also developed the Backward Steering heuristic algorithm to solve the problem. KangMT1999 [53] explored parallel-machine batch scheduling with SDST in a multi-period production system; they developed a solution method based on column generation and branch-and-bound to minimise the difference between operational costs and sales revenue under the condition that the demand for each product in each period does not exceed the production capacity. They also tested two heuristics to speed up the multi-period scheduling.

Meyr2002 [54] extended their study of the general lot-sizing and scheduling problem for parallel production lines to minimise production, inventory, and SDST-related costs. They developed a hybrid solution method to solve a real-world problem in the consumer goods industry. Clark2003 [55] proposed three MIP models for batch scheduling with capacity constraints and SDSTs to optimise the Master Production Schedule (MPS) and Material Requirements Planning (MRP). Considering the high computational complexity of these models, two approximation algorithms were developed to reduce the number of binary variables, and a heuristic method called Relax-and-Fix was used to decompose the original model into a series of smaller sub-problems. DearaujoAC2007 [56] proposed an effective solution methodology using MIP approaches and heuristics for integrated lot-sizing and scheduling problems, considering late orders and SDST. They employed the Relax-and-Fix method and performed computational tests with a high-performance MIP solver. Additionally, the authors developed three local search methods to improve the performance of the Relax-and-Fix method.

FerreiraMR2009 [57] proposed an MIP model for integrated production lot-sizing and scheduling decisions in the beverage industry. Their study focused on production bottlenecks and inter-stage synchronization requirements. The authors proposed a Relax-and-Fix-based solution method and explored different variable splitting and fixing strategies for solving the problem. TosoMC2009 [58] explored a complex production scheduling problem that involves multiple products and SDSTs for a two-stage product structure. They proposed an integrated MIP formulation including inventory and lot size variables and approached the optimisation objective based on the Transportation

Problem (TP). ClarkMT2010 [59] proposed a new approach for combined production lot-sizing and scheduling based on the asymmetric TSP and tested its application in the animal nutrition production industry. They compared different MILP models and solution methods, highlighting the advantages of the asymmetric TSP model in solving the problem of minimising inventory, delay, and overtime costs while considering practical constraints such as capacity, demand, and minimum batch size.

TranschelMKLE2011 [60] researched joint lot-sizing and scheduling for production with a two-stage product structure, considering SDSTs. The authors proposed a hybrid MIP formulation based on the Quantity-based Transportation Problem (QTP) and the Proportion-based Transportation Problem (PTP). Their model addresses practical constraints such as minimum production quantities between products while minimising operational costs. SeeannerM2013 [61] proposed an MIP model that takes into account production factors such as capacity, SDSTs, holding costs, external procurement, overtime, and standby to minimise total costs in the integrated problem of lot-sizing and scheduling. Their study proposed a general time structure that can flexibly avoid lead time between different production stages and allow for splitting production batches and setup time within each scheduling period to improve efficiency. SeeannerAM2013 [62] continued the development path of integrated lot-sizing and scheduling in a multi-level production system. They developed a solution method that combines Variable Neighborhood Decomposition Search (VNDS) and Fix & Optimise. The developed scheme demonstrated high flexibility and effectiveness in solving complex scheduling problems, such as the general lot-sizing and scheduling problem for multiple production stages.

Overall, the studies under this branch not only take into account the complexity of production schedules but also consider various factors in real-life production environments. From the early studies on single-machine multi-product production scheduling to the later multi-stage batch scheduling, this branch has contributed to the developments in FSSPs. Continuing from the last node of this branch, SioudG2018 [63] developed a metaheuristic for solving permutation FSSP with SDSTs to minimise the makespan.

4.4. Branch IV

The literature on this path includes publications between 2009 and 2018; this path branches out from NaderiZR2009 [25] (hybrid flow shops) in the global main path to NaderiZS2009 [64] (flexible flow shops) and continues to LiYRCS2018 [65] (no-wait flow shop). This branch will later merge into the global main path through HuangPG2020 [32], which explores distributed FSSPs.

NaderiZS2009 [64] adjusted the Electromagnetism Algorithm (EMA) to solve the flexible FSSP with SDSTs and independent transportation times, assuming multiple transportation tools for job delivery. The goal was to minimise the total weighted tardiness to enhance production efficiency and reduce delays. The problem was mathematically modelled using MILP and solved using EMA for large-scale instances. JolaiRA2012 [66] studied the no-wait version of the flexible FSSP intending to minimise the makespan; they developed the population-based SA, an adapted version of the Imperialist Competitive Algorithm (ICA), and a hybrid version of the two to solve this problem. SamarghandiE2014 [67] extended the Particle Swarm Optimisation (PSO) algorithm to solve the no-wait FSSP-SDST considering the makespan. They introduced the matrix coding mechanism that enables the solution algorithm to handle SDSTs more effectively. NaganDN2014 [68] studied the no-wait variant of FSSP-SDST using a hybrid of the GA and Evolutionary Clustering Search (ECS) algorithms equipped with a local search and considered the makespan as the optimisation goal.

Similarly, NaganoMA2015 [69] focused on the no-wait FSSP-SDST and introduced a new constructive heuristic, called QUARTS, to

minimise the total completion time of all jobs. This problem has applications in the metal, plastics, and chemical industries. Their method identifies the best job sequence by decomposing the problem into a combination of tasks. LiYRCS2018 [65] studied the no-wait FSSP-SDST while considering the learning effect; they established a position-based model for the learning effect and employed it in an accelerated construction heuristic method embedded into the IG algorithm to minimise the total process time. This research provided new best-found solutions to this problem and suggested new ideas for future research.

In general, these studies demonstrate advances in solving the no-wait FSSP-SDST and its variants. From initial constructive heuristics to hybrid metaheuristics, which take into account more complex factors that reflect real-world situations, such as family setup times and learning effects.

4.5. Branch V

The path starting from Gomez-gasquetAL2012 [70] and finishing at SioudG2018 [63] has branched from NaderiRZ2009 [26] in the global main path, which studied the hybrid flexible FSSP with SDSTs. This branch later converges into the global main path through HuangPG2020 [32], which studied the distributed permutation FSSP with SDSTs. Although the studies from this branch are focused on FSSP with SDSTs, their focus differs.

Gomez-gasquetAL2012 [70] studied hybrid FSSP with SDSTs to minimise the makespan; they improved the GA with an agent-based method, which utilises the characteristics of software agents. In their method, new individuals are generated based on local competition rather than global competition, which enables the parameters to be dynamically adjusted according to changes in the environment and facilitates the application of customized genetic operators.

Studying FSSPs with SDSTs, VanchipuraSB2014 [71] introduced new constructive heuristics based on Variable Neighborhood Descent (VND) to minimise the makespan. Their experiments showed that the VND method can considerably improve the solution quality. Finally, SioudG2018 [63] focused on permutation FSSP with SDST and developed a new heuristic based on setup times and the Enhanced Migrating Birds Optimisation (EMBO) to minimise the makespan. A variety of improved computational mechanisms were embedded into EMBO to show the competitiveness of EMBO for combinatorial optimisation.

4.6. Branch VI

Rooting from PanGLG2017 [30] on the global main path, which worked on hybrid FSSP with SDSTs, MengZSZRL2020 [72] began this branch, and LiLGZPTM2021 [73] is its closing point. This branch converges into the global main path through MengP2021 [33], which explores the distributed permutation FSSP with SDSTs and heterogeneous factories. This path branched off again from MengP2021 [33], resulting in a brief but important path through GuoSZML2022 [74]. The return to the main path took place through HanHZQLLG2022 [35], which is focused on the distributed blocking FSSP with SDSTs.

MengZSZRL2020 [72] studied hybrid FSSP with SDSTs. They proposed eight MILP models where the makespan is the optimisation objective. The extended formulations account for no-wait and blocking settings for unrelated parallel machines. They also conducted numerical experiments using CPLEX to evaluate the computational complexity of the models. LiLGZPTM2021 [73] explored the distributed hybrid FSSP with SDSTs while involving multiple factories, multiple stages, and multiple parallel machines; the modified discrete ABC algorithm was used to minimise the makespan in the proposed model. In their approach, an innovative machine position-based mathematical model was showcased to facilitate the encoding and decoding methods of the solution algorithm. MengP2021 [33] investigated the distributed permutation FSSP with SDSTs and heterogeneous factories, receiving a new extension into the ABC algorithm with the goal of minimising the

makespan. GuoSZML2022 [74] considered the distributed permutation FSSP with SDSTs and proposed a new metaheuristic algorithm based on a differential flight strategy and the known Discrete Fruit Fly Optimisation (DFFO_DF) algorithm. They introduced four new perturbation mechanisms and a differential flight strategy to enhance the exploration power of the solution algorithm.

As discussed in Section 4.3, HanHZQLLG2022 [35] studied distributed blocking FSSP with SDSTs to minimise energy consumption costs. The big picture shows a shift from FSSPs to hybrid FSSPs, distributed FSSPs, and finally, flow shop GSPs. The development of flow shop GSP has its roots in the study of HanHZQLLG2022, where QinHWLLP2022 [36] inspired the latest articles (sink nodes) published in 2023 (SekkalB2023 [37], WangHWLGL2023 [38], LiHZWLG2023 [39], WangHPLW2023 [40]). The recent focus is on GSPs, considering the characteristics of SDSTs along with factors such as the learning effect and energy consumption. The need for solutions that balance financial and non-financial optimisation objectives reflects the industry's demand for carbon neutrality in addition to improving efficiency.

Overall, the development branches of production scheduling with SDSTs have progressed to solving more complex scheduling problems, particularly GSP, and problems that include interrelated sub-problems, such as plant allocation. Considering the developments in the depth of the scheduling problems, researchers have focused on dealing with complexities and multi-variable production scheduling through a combination of different model-building methods and optimisation strategies. This includes considering factors such as batch processing, variable sub-batches, and SDSTs, as well as developing new algorithms to improve the efficiency and effectiveness of solutions. The research at this stage not only demonstrates innovation in algorithm and model development but also reflects a focus on bridging the gap between scheduling theory and practice.

5. Discussions based on cluster analysis

A total of 1973 literature items were imported into MainPath480 for keyword analysis. The GroupFinder function was used for clustering based on the Edge-Betweenness Clustering algorithm. Results identified a total of 22 clusters, of which the top ten clusters are considered for in-depth analysis. These clusters include 441, 346, 197, 166, 97, 94, 84, 39, 29, and 27 documents, respectively. Table 1 summarizes the main features of the top clusters. In this table, the word clouds highlight the centrality of the keywords used under each theme. In addition, a trend diagram is presented, depicting the expected future trend based on past development for each research theme. The research cluster on flexible JSSPs began the growth phase later than the other clusters, and hence, it is expected to attract relatively more attention concerning SDST-related issues. Fig. 5 shows a summary of the historical trend chart of the top ten clusters.

The key-extension main path with a key-route value of 5 is now considered for identifying the main path under each research theme based on this, the most recent development (blue nodes in Fig. 6) is reviewed.

Cluster 1. The first cluster focuses on solving FSSPs with SDSTs and other constraints, mostly aiming to minimise the makespan as the main objective function. This cluster essentially corresponds to the global main path, except that the main path diverges in LiHZWLG2023 [39]. This article discussed a hybrid FSSP that considers batch processing, variable sub-batches, SDSTs, and transportation time. The research under this cluster was mainly focused on a single goal in the early stages but gradually evolved into more complex optimisation with conflicting objectives. As the research under this cluster deepened, the scope of the problem expanded to include time windows, energy costs, and transportation time. A variety of advanced algorithms, such as GAs, ABC, MOSA, and IG algorithms, were used. Most recently, integrating multiple methods and strategies has drawn attention to the improvement of the performance and adaptability of solution algorithms.

Cluster 2. The core of the research in this cluster is PMSP, and to solve the problems, researchers have mainly developed heuristic and metaheuristic algorithms. Makespan is the most commonly used optimisation criterion. As the sink node of the development path under this cluster, Chen2023 [75] studied an order acceptance-based scheduling problem on unrelated parallel machines to select the most gainful orders and arrange them on machines to maximise profitability. They developed a population-based IG algorithm to solve the problem, which was integrated with a neighbourhood descent search mechanism to improve the selected solution.

Cluster 3. The core of this cluster is formed around lot-sizing-based scheduling. The development path of the third cluster has resulted in four sink nodes. MaccawleyMPT2022 [76] explored setup time in scheduling applications in the winery industry for the planning of bottling lines to reduce inventory costs, increase production efficiency, and improve customer fulfillment. They proposed an MIP formulation for the batch scheduling problem and a new decomposition-based solution algorithm to solve it. AlimianGTR2022 [77] integrated preventive maintenance variables into scheduling with setup times and employed time-based maintenance and age reduction modelling to determine production line availability, thereby reducing the impact of sudden failures and corrective maintenance. They proposed an MILP model and a rolling horizon algorithm to solve it. RohaninejadH2023 [78] explored the problem of multi-level lot-sizing and JSSPs with SDSTs, lot streaming, and machine capacity constraints. They introduced a novel formulation and two efficient heuristics to solve the problem. Finally, LeeL2023 [79] proposed new inequalities and extended optimisation models for the integrated lot-sizing and scheduling with setup times to minimise inventory, stockout, production, and setup costs while satisfying dynamic demand.

Cluster 4. The single-machine scheduling problem is at the core of the fourth cluster. This cluster has led to two recent developments by DefreitasF2022 [80] and CostaF2022 [81]. DefreitasF2022 [80] proposed a mathematical model based on TSP for production scheduling in a single-machine environment while considering SDSTs. They suggested using the results obtained by the heuristic algorithms as an initial solution to improve the approximation process and demonstrated that the approach is better than the baseline, both in terms of solution quality and computational time. CostaF2022 [81] explored the single-machine scheduling problem with SDSTs while considering variable and flexible maintenance, as well as dynamic job arrival—a situation that is prevalent in semiconductor manufacturing. They aimed to minimise the total delay time as the objective function, for which an improved harmony search algorithm was developed.

Cluster 5. This research theme focuses on group technology and addresses batch and flow shop scheduling issues. Four of the articles under this research theme form the main development path in the SDST scheduling literature. The studies of SekkalB2023 [28] and WangHPLW2023 [31] constitute the most recent developments under this cluster. These are reviewed in Section 4.4 since they also appeared on the main path. This concurrence highlights the importance of this cluster as an ongoing research theme.

Cluster 6. This cluster focuses on the scheduling issues related to learning effects and deterioration. FengHSW2023 [82] explored the shop-floor feature where the processing time of a job changes with its position in the sequence, known as the learning effect. This study assumed common, loose, and restricted lead time allocations to minimise the weighted sum of sequence-dependent weights and lead time allocation costs.

Cluster 7. This cluster mainly focuses on the flexible JSSP using various solution algorithms and objective functions and incorporating practical constraints into the optimisation methods. Four recent developments have resulted from the main path under this cluster. SongL2022 [83] discussed the flexible JSSP with SDSTs. They proposed a mathematical model to minimise the makespan by employing an improved version of the Quantum Cat Swarm Optimisation (QCSO)

Table 1 (continued)

| C3: Batch scheduling problems (197 articles) | | C4: Single-machine scheduling problems (170 articles) | |
|---|------|--|------|
| Lot Sizing | 0.47 | Single Machine Scheduling | 0.32 |
| Heuristic Algorithm | 0.11 | Tardiness | 0.18 |
| Mixed Integer Programming | 0.09 | Heuristic Algorithm | 0.12 |
| Genetic Algorithm | 0.06 | Genetic Algorithm | 0.09 |
| Mixed Integer Linear Programming | 0.06 | Branch and Bound | 0.09 |
| Mathematical Programming | 0.05 | Meta-Heuristic | 0.08 |

Growth of Papers

| C5: Group scheduling problems (97 articles) | | C6: Learning and deterioration effects in scheduling (94 articles) | |
|---|------|--|------|
| Group Technology | 0.32 | Learning Effect | 0.61 |
| Genetic Algorithm | 0.16 | Single Machine Scheduling | 0.58 |
| Flow Shop | 0.13 | Deterioration | 0.29 |
| Heuristic Algorithm | 0.11 | Resource Allocation | 0.06 |
| Lower Bound | 0.11 | Bi-criteria | 0.05 |
| Batch Scheduling | 0.09 | Branch and Bound | 0.05 |

Growth of Papers

Growth of Papers

(continued on next page)

Table 1 (continued)

| C7: Flexible job shop scheduling problems (84 articles) | | C8: Dynamic job shop scheduling problems (39 articles) | |
|---|------|--|------|
| Flexible Job Shop | 0.44 | Job Shop | 0.36 |
| Genetic Algorithm | 0.17 | Dynamic Programming | 0.13 |
| Job Shop | 0.13 | Genetic algorithm | 0.10 |
| Bi-criteria | 0.08 | Heuristic Algorithm | 0.10 |
| Heuristic Algorithm | 0.06 | Branch and Bound | 0.08 |
| Makespan | 0.06 | Flow Shop | 0.08 |
|  | |  | |
|  | |  | |
| C9: Manufacturing system issues (29 articles) | | C10: Order acceptance and scheduling (27 articles) | |
| Throughput | 0.17 | Order Acceptance | 0.65 |
| Manufacturing Systems | 0.14 | Single Machine Scheduling | 0.23 |
| Production Scheduling | 0.14 | Meta-Heuristic | 0.19 |
| Multiple Products | 0.10 | Heuristic Algorithm | 0.15 |
| Stochastic Demand | 0.10 | Make-to-Order | 0.08 |
| Bernoulli Reliability Model | 0.07 | Mathematical Formulation | 0.08 |
|  | |  | |
|  | |  | |

algorithm, which performed better than GAs. ParkHP2023 [84] introduced a case-based reasoning framework for application in semiconductor packaging facilities. Their model accounted for practical features of initial setup status, available machines, production needs, job waiting time, machine downtime, and potential changes in their relative importance. The contribution of this paper was to propose a novel two-stage framework that uses case-based reasoning for efficient scheduling. MengZGD2023 [85] studied the energy-aware flexible JSSP considering SDSTs and transportation time. They proposed an MILP formulation for minimising energy consumption in processing, maintenance, idle time, and transportation activities and introduced a shutdown/startup strategy to reduce idle energy consumption. Finally, YanZSZ2023 [86] explored the multi-objective flexible JSSP while simultaneously accounting for SDSTs, transportation, and delivery time. They improved the Ant Colony Optimisation (ACO) algorithm to solve this problem, considering the makespan, total workload, workload on key machines, and early/tardy penalties.

Cluster 8. Cluster eight is formed around classic JSSPs with SDSTs. This cluster experienced a relatively early closure, with no significant developments after 2014. The last seminal study under this cluster, Shen2014 [87] discussed the JSSP with SDSTs and extended the TS algorithm to minimise the makespan.

Cluster 9. With a total of 10 documents, cluster nine includes studies that explore practical issues related to manufacturing systems, particularly those related to inventory, multiple products, and throughput. WangDWJ2022 [88] offered knowledge on transient analysis based on the Markov chain model to study the performance of the series production line, considering the makespan and total processing time.

Cluster 10. This cluster predominantly concerns Order Acceptance Scheduling (OAS), especially in Make-to-order production environments. Much of the research under this category utilizes single-machine and parallel-machine scheduling. The main path of cluster 10 has concluded with four recent contributions. BruniKD2020 [89] studied a PMSP in which the acceptance or rejection of jobs, as well as the

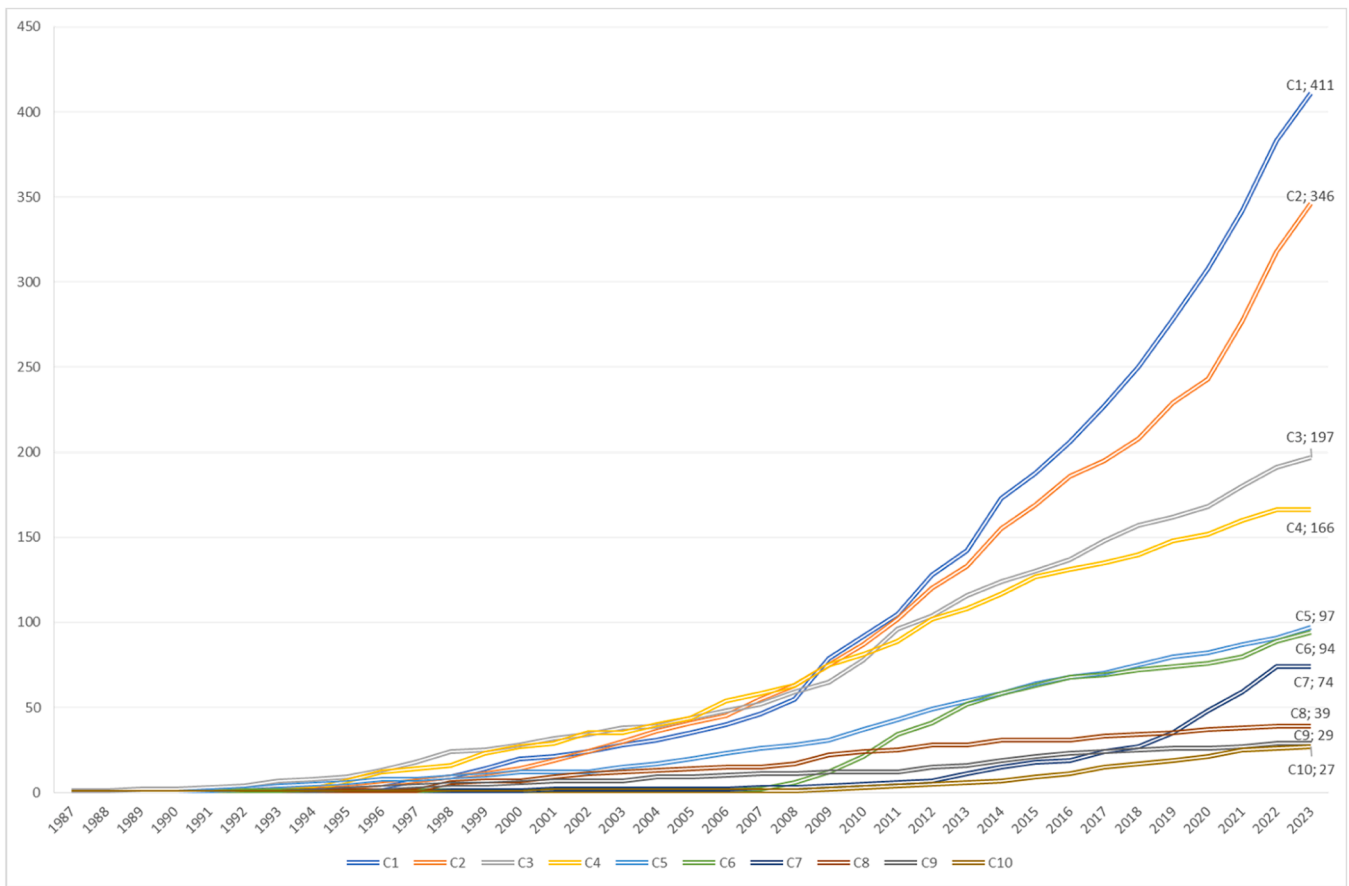


Fig. 5. Publication records of the top ten clusters.

processing time and SDSTs, are stochastic. They developed a hybrid metaheuristic based on neighbourhood search to minimise risk, which is measured using distributional robust optimisation and conditional value-at-risk. BicakcDK2021 [90] focused on OAS considering SDSTs, release dates, and deadlines. They proposed a new mathematical model and compared it with several baselines. They also developed a new heuristic based on VNS and SA to solve large-scale instances. TarhanO2022 [91] explored a generalization of the OAS problem and proposed MILP and constraint programming formulations to model it. They developed an advanced solution method that combined the so-called Time-Bucket Model with VNS and TS algorithms. Most recently, MahmoudnazlouANE2023 [92] studied OAS with time windows and SDSTs and developed a hybrid of ICA and SA algorithms to solve the problem for large instances. They also found a new upper bound for the problem by relaxing one of the constraints in the MILP model.

6. Concluding remarks

6.1. Conclusions

This systematic review explored the development of knowledge of scheduling problems with SDSTs. Over 2100 articles published between 1986 and 2024 were considered. First, the global main path was identified using the MPA method. Based on this, the seminal articles contributing to the advances of SDST in production scheduling were reviewed. The key extensions of the main path were then considered to discover the secondary research interests in the field. A total of 22 clusters were identified using the CA method. Based on these clusters, an overview of the research objectives, methods, and practicality branches of SDSTs was explored.

Overall, the KDT spans four stages. Research from 1992 to 2008 primarily focused on FSSPs; articles between 2009 and 2019 concentrated on hybrid FSSPs. From 2020 to 2022, the distributed permutation flow shop became the dominant research focus, and the most recent studies up to the end of 2023 primarily explored GSPs. The key extensions from the main path involved extending flow shop systems to hybrid, distributed, and group scheduling contexts. Moreover, solutions algorithms have gradually shifted towards hybridization. The CA method identified ten main clusters: FSSP with makespan optimisation, PMSP, batch scheduling problems, single-machine scheduling problems, GSPs, learning and deterioration effects in scheduling, flexible JSSP, dynamic JSSP, manufacturing system issues, and order acceptance integrated with scheduling. Except for the cluster associated with single-machine scheduling, which will soon experience saturation, the others are projected to maintain their growth pace until mid-2035.

6.2. Directions for future research

Optimisation objective. SDST has been extensively studied in parallel machines, flow shops, and job shops. The scheduling problems involving SDSTs have primarily focused on minimizing the makespan, total delay, and total completion time. There is a need for further exploration of multi-objective scheduling that considers SDSTs and non-financial objectives, which helps provide additional insights into system efficiency beyond cost and time optimisation goals. Several recent studies have addressed multi-objective optimisation that considers energy efficiency, renewable energy constraints, and energy consumption, which opens a new direction for future research.

Production setting. While SDSTs have been well-studied in more conventional environments like flow shops and job shops, there is considerable potential to develop more complex models for SDSTs in

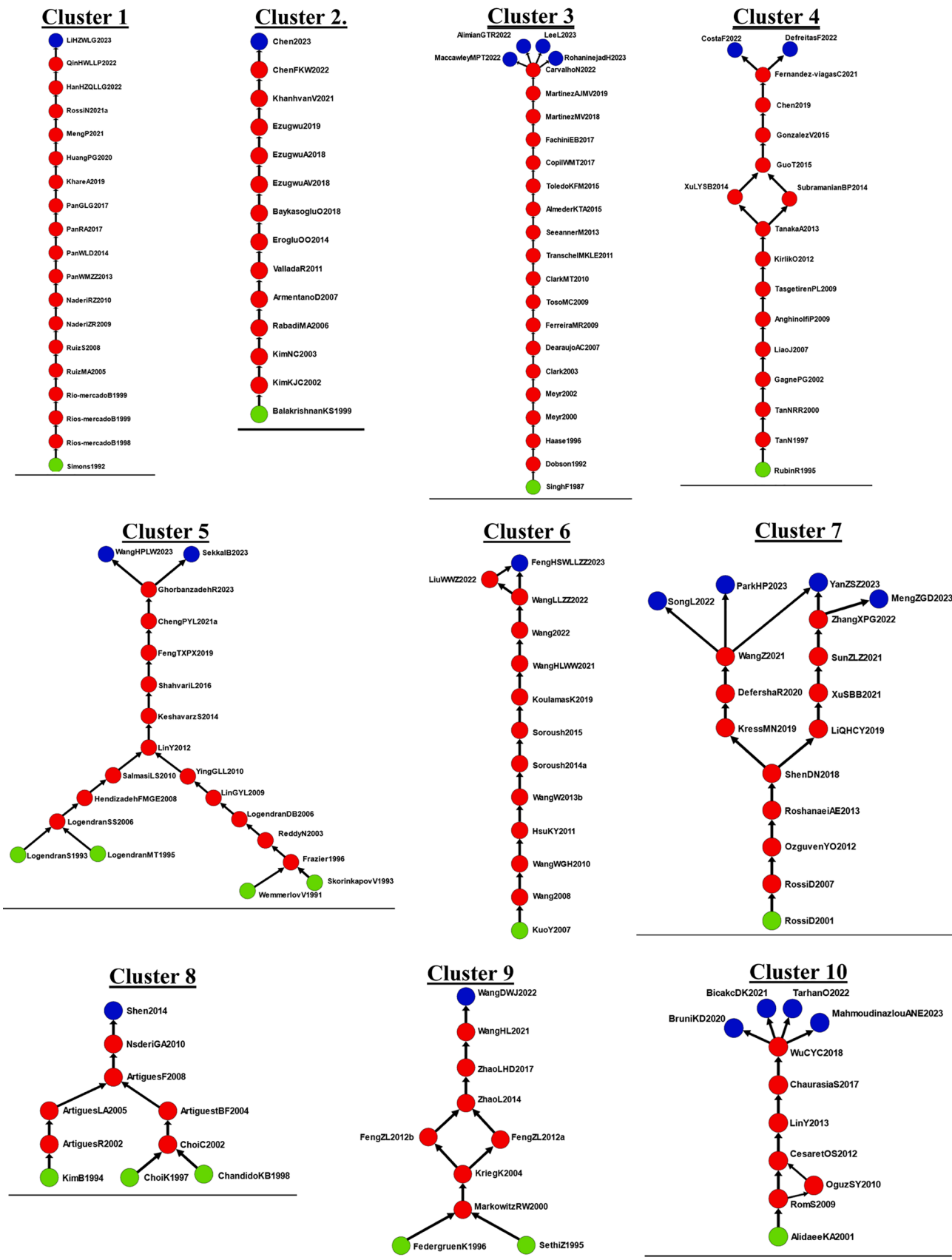


Fig. 6. The main path leading to the most recent development in the top clusters.

open shop environments. Moreover, further research may explore the application of SDSTs in high-volume manufacturing industries to validate their impact and refine the models as needed. Practical implementations in these sectors will enhance our understanding of the applicability and scalability of SDSTs in different production settings.

Digital technologies. Exploring the current development trajectories of SDST confirms a growing trend for integrating emerging digital technologies, such as machine learning and big-data analysis, into optimisation approaches. This may help reduce avoidable setups and optimise operations that are often intangible and non-value-adding. The incorporation of such techniques could further improve the effectiveness and efficiency of scheduling systems beyond the current optimality norms and offer novel solutions to existing challenges in SDSTs.

Emerging challenges. There are technical challenges in incorporating SDSTs with the new industrial landscapes, notably Industry 5.0 and cyber-physical systems. A broader discussion is required to highlight the relevance of SDSTs within the digital transformation context. Additive manufacturing is another arena where SDST requires new developments. Finally, understanding the impact of setups on automated manufacturing environments and human-robot collaborations will be critical to advancing the field and ensuring that modern demands are well addressed.

6.3. Limitations of the existing review

While this study utilized MPA and CA to identify broad patterns of knowledge development, it did not provide the granularity needed for detailed analyses of specific research subgroups and studies. In particular, research areas such as SDST scheduling in different production environments require a more in-depth examination of their unique challenges, methodologies, and advancements. Due to space constraints and the methodological nature of MPA and CA, our analysis primarily focuses on overarching trends rather than providing a comprehensive review of each subgroup. Future research may address this limitation by employing more targeted review methodologies, such as systematic literature reviews or bibliometric analyses that focus on individual subfields. These approaches would enable a deeper exploration of specific research areas, helping to uncover nuanced insights and identify critical gaps within each segment of the literature.

Due to the extensive volume of the literature analysed—totaling 2195 articles—conducting a thorough analysis and drafting the manuscript was time-consuming. Additionally, as the statistical analysis was based on annual publications, it was not feasible to include the most recent literature beyond 2023 in this study. Future research may address this limitation by conducting the MPA every 5–10 years to periodically update the research landscape related to SDSTs and incorporate the latest advancements in the field.

CRedit authorship contribution statement

Kuo-Ching Ying: Writing – review & editing, Validation, Supervision, Software, Project administration, Methodology, Conceptualization. **Pourya Pourhejazy:** Writing – original draft, Investigation, Formal analysis. **Zhi-Rong Lin:** Visualization, Methodology, Formal analysis, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

- [1] Cheng CY, Pourhejazy P, Ying KC, Lin CF. Unsupervised learning-based artificial Bee colony for minimizing non-value-adding operations. *Appl Soft Comput* 2021; 105:107280. <https://doi.org/10.1016/j.asoc.2021.107280>.
- [2] Pourhejazy P. Production management and supply chain integration. the palgrave handbook of supply chain management. Cham: Springer International Publishing; 2024. p. 315–40. https://doi.org/10.1007/978-3-031-19884-7_86.
- [3] Gao K, Huang Y, Sadollah A, Wang L. A review of energy-efficient scheduling in intelligent production systems. *Complex Intell Syst* 2020;6:237–49. <https://doi.org/10.1007/s40747-019-00122-6>.
- [4] Le Hesran C, Ladier A-L, Botta-Genoulaz V, Laforest V. Operations scheduling for waste minimization: a review. *J Clean Prod* 2019;206:211–26. <https://doi.org/10.1016/j.jclepro.2018.09.136>.
- [5] Allahverdi A, Ng CT, Cheng TCE, Kovalyov MY. A survey of scheduling problems with setup times or costs. *Eur J Oper Res* 2008;187:985–1032. <https://doi.org/10.1016/j.ejor.2006.06.060>.
- [6] Allahverdi A, Gupta JND, Aldowaisan T. A review of scheduling research involving setup considerations. *Omega (Westport)* 1999;27:219–39. [https://doi.org/10.1016/S0305-0483\(98\)00042-5](https://doi.org/10.1016/S0305-0483(98)00042-5).
- [7] Chen F, Luo C, Gong W, Lu C. Two-stage adaptive memetic algorithm with surprisingly popular mechanism for energy-aware distributed hybrid flow shop scheduling problem with sequence-dependent setup time. *Complex Syst Model Simul* 2024;4:82–108. <https://doi.org/10.23919/CSMS.2024.0003>.
- [8] Qiu J, Liu J, Peng C, Chen Q. A novel predictive-reactive scheduling method for parallel batch processor lot-sizing and scheduling with sequence-dependent setup time. *Comput Ind Eng* 2024;189:109985. <https://doi.org/10.1016/j.cie.2024.109985>.
- [9] Zheng Q, Zhang Y, Tian H, He L. A cooperative adaptive genetic algorithm for reentrant hybrid flow shop scheduling with sequence-dependent setup time and limited buffers. *Complex Intell Syst* 2024;10:781–809. <https://doi.org/10.1007/s40747-023-01147-8>.
- [10] Barak S, Javanmard S, Moghdani R. Dual resource constrained flexible job shop scheduling with sequence-dependent setup time. *Expert Syst* 2024;41. <https://doi.org/10.1111/exsy.13669>.
- [11] Pourhejazy P, Cheng C-Y, Ying K-C, Nam NH. Meta-Lamarckian-based iterated greedy for optimizing distributed two-stage assembly flowshops with mixed setups. *Ann Oper Res* 2022. <https://doi.org/10.1007/s10479-022-04537-2>.
- [12] Liu F, Gao K, Li D, Sadollah A. Ensemble evolutionary algorithms equipped with Q-learning strategy for solving distributed heterogeneous permutation flowshop scheduling problems considering sequence-dependent setup time. *IET Collab Intell Manuf* 2024;6. <https://doi.org/10.1049/cim2.12099>.
- [13] Shao Z, Shao W, Chen J, Pi D. A feedback learning-based selection hyper-heuristic for distributed heterogeneous hybrid blocking flow-shop scheduling problem with flexible assembly and setup time. *Eng Appl Artif Intell* 2024;131:107818. <https://doi.org/10.1016/j.engappai.2023.107818>.
- [14] Allahverdi A. The third comprehensive survey on scheduling problems with setup times/costs. *Eur J Oper Res* 2015;246:345–78. <https://doi.org/10.1016/j.ejor.2015.04.004>.
- [15] Sharma P, Jain A. A review on job shop scheduling with setup times. *Proc Inst Mech Eng B J Eng Manuf* 2016;230:517–33. <https://doi.org/10.1177/0954405414560617>.
- [16] Ying K-C, Pourhejazy P, Huang X-Y. Revisiting the development trajectory of parallel machine scheduling. *Comput Oper Res* 2024;168:106709. <https://doi.org/10.1016/j.cor.2024.106709>.
- [17] Liu JS, Lu LYY, Ho MH-C. A few notes on main path analysis. *Scientometrics* 2019; 119:379–91. <https://doi.org/10.1007/s11192-019-03034-x>.
- [18] Girvan M, Newman MEJ. Community structure in social and biological networks. *Proc Natl Acad Sci* 2002;99:7821–6. <https://doi.org/10.1073/pnas.122653799>.
- [19] Simons J. Heuristics in flow shop scheduling with sequence dependent setup times. *Omega (Westport)* 1992;20:215–25. [https://doi.org/10.1016/0305-0483\(92\)90075-1](https://doi.org/10.1016/0305-0483(92)90075-1).
- [20] Ríos-Mercado RZ, Bard JF. Heuristics for the flow line problem with setup costs. *Eur J Oper Res* 1998;110:76–98. [https://doi.org/10.1016/S0377-2217\(97\)00213-0](https://doi.org/10.1016/S0377-2217(97)00213-0).
- [21] Ríos-Mercado RZ, Bard JF. An enhanced TSP-based heuristic for makespan minimization in a flow shop with setup times. *J Heurist* 1999;5:53–70. <https://doi.org/10.1023/A:1009691028143>.
- [22] Ríos-Mercado RZ, Bard JF. A branch-and-bound algorithm for permutation flow shops with sequence-dependent setup times. *IEE Trans* 1999;31:721–31. <https://doi.org/10.1080/07408179908969871>.
- [23] Ruiz R, Maroto C, Alcaraz J. Solving the flowshop scheduling problem with sequence dependent setup times using advanced metaheuristics. *Eur J Oper Res* 2005;165:34–54. <https://doi.org/10.1016/j.ejor.2004.01.022>.
- [24] Ruiz R, Stützle T. An Iterated greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. *Eur J Oper Res* 2008;187:1143–59. <https://doi.org/10.1016/j.ejor.2006.07.029>.
- [25] Naderi B, Zandieh M, Roshanaei V. Scheduling hybrid flowshops with sequence dependent setup times to minimize makespan and maximum tardiness. *Int J Adv Manuf Technol* 2009;41:1186–98. <https://doi.org/10.1007/s00170-008-1569-3>.
- [26] Naderi B, Ruiz R, Zandieh M. Algorithms for a realistic variant of flowshop scheduling. *Comput Oper Res* 2010;37:236–46. <https://doi.org/10.1016/j.cor.2009.04.017>.
- [27] Pan Q, Wang L, Mao K, Zhao J, Zhang M. An effective artificial bee colony algorithm for a real-world hybrid flowshop problem in steelmaking process. *IEEE*

- Trans Autom Sci Eng 2013;10:307–22. <https://doi.org/10.1109/TASE.2012.2204874>.
- [28] Pan Q-K, Wang L, Li J-Q, Duan J-H. A novel discrete artificial bee colony algorithm for the hybrid flowshop scheduling problem with makespan minimisation. *Omega* (Westport) 2014;45:42–56. <https://doi.org/10.1016/j.omega.2013.12.004>.
- [29] Pan Q-K, Ruiz R, Alfaro-Fernández P. Iterated search methods for earliness and tardiness minimization in hybrid flowshops with due windows. *Comput Oper Res* 2017;80:50–60. <https://doi.org/10.1016/j.cor.2016.11.022>.
- [30] Pan Q-K, Gao L, Li X-Y, Gao K-Z. Effective metaheuristics for scheduling a hybrid flowshop with sequence-dependent setup times. *Appl Math Comput* 2017;303: 89–112. <https://doi.org/10.1016/j.amc.2017.01.004>.
- [31] Khare A, Agrawal S. Scheduling hybrid flowshop with sequence-dependent setup times and due windows to minimize total weighted earliness and tardiness. *Comput Ind Eng* 2019;135:780–92. <https://doi.org/10.1016/j.cie.2019.06.057>.
- [32] Huang J-P, Pan Q-K, Gao L. An effective iterated greedy method for the distributed permutation flowshop scheduling problem with sequence-dependent setup times. *Swarm Evol Comput* 2020;59:100742. <https://doi.org/10.1016/j.swevo.2020.100742>.
- [33] Meng T, Pan Q-K. A distributed heterogeneous permutation flowshop scheduling problem with lot-streaming and carryover sequence-dependent setup time. *Swarm Evol Comput* 2021;60:100804. <https://doi.org/10.1016/j.swevo.2020.100804>.
- [34] Rossi FL, Nagano MS. Heuristics and iterated greedy algorithms for the distributed mixed no-idle flowshop with sequence-dependent setup times. *Comput Ind Eng* 2021;157:107337. <https://doi.org/10.1016/j.cie.2021.107337>.
- [35] Han X, Han Y, Zhang B, Qin H, Li J, Liu Y, et al. An effective iterative greedy algorithm for distributed blocking flowshop scheduling problem with balanced energy costs criterion. *Appl Soft Comput* 2022;129:109502. <https://doi.org/10.1016/j.asoc.2022.109502>.
- [36] Qin H, Han Y, Wang Y, Liu Y, Li J, Pan Q. Intelligent optimization under blocking constraints: a novel iterated greedy algorithm for the hybrid flow shop group scheduling problem. *Knowl Based Syst* 2022;258:109962. <https://doi.org/10.1016/j.knsys.2022.109962>.
- [37] Sekkal DN, Belkaid F. A multi-objective optimization algorithm for flow shop group scheduling problem with sequence dependent setup time and worker learning. *Expert Syst Appl* 2023;233:120878. <https://doi.org/10.1016/j.eswa.2023.120878>.
- [38] Wang Y, Han Y, Wang Y, Li J, Gao K, Liu Y. An effective two-stage iterated greedy algorithm for distributed flowshop group scheduling problem with setup time. *Expert Syst Appl* 2023;233:120909. <https://doi.org/10.1016/j.eswa.2023.120909>.
- [39] Li C, Han Y, Zhang B, Wang Y, Li J, Gao K. A novel collaborative iterative greedy algorithm for hybrid flowshop scheduling problem with batch processing machines and variable sublots. *Int J Prod Res* 2023;1–21. <https://doi.org/10.1080/00207543.2023.2253925>.
- [40] Wang Y, Han Y, Pan Q, Li H, Wang Y. Redefining hybrid flow shop group scheduling: unveiling a novel hybrid modeling paradigm and assessing 48 MILP and CP models. *Swarm Evol Comput* 2023;83:101416. <https://doi.org/10.1016/j.swevo.2023.101416>.
- [41] Singh H, Foster JB. Production scheduling with sequence dependent setup costs. *IEE Trans* 1987;19:43–9. <https://doi.org/10.1080/07408178708975368>.
- [42] Ovacik IM, Uzsoy R. Worst-case error bounds for parallel machine scheduling problems with bounded sequence-dependent setup times. *Oper Res Lett* 1993;14: 251–6. [https://doi.org/10.1016/0167-6377\(93\)90089-Y](https://doi.org/10.1016/0167-6377(93)90089-Y).
- [43] Sivrikaya-Şerifoğlu F, Ulusoy G. Parallel machine scheduling with earliness and tardiness penalties. *Comput Oper Res* 1999;26:773–87. [https://doi.org/10.1016/S0305-0548\(98\)00090-2](https://doi.org/10.1016/S0305-0548(98)00090-2).
- [44] França PM, Gendreau M, Laporte G, Müller FM. A tabu search heuristic for the multiprocessor scheduling problem with sequence dependent setup times. *Int J Prod Econ* 1996;43:79–89. [https://doi.org/10.1016/0925-5273\(96\)00031-X](https://doi.org/10.1016/0925-5273(96)00031-X).
- [45] Kurz ME, Askin RG. Heuristic scheduling of parallel machines with sequence-dependent set-up times. *Int J Prod Res* 2001;39:3747–69. <https://doi.org/10.1080/00207540110064938>.
- [46] Kurz ME, Askin RG. Comparing scheduling rules for flexible flow lines. *Int J Prod Econ* 2003;85:371–88. [https://doi.org/10.1016/S0925-5273\(03\)00123-3](https://doi.org/10.1016/S0925-5273(03)00123-3).
- [47] Ruiz R, Maroto C. A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. *Eur J Oper Res* 2006;169:781–800. <https://doi.org/10.1016/j.ejor.2004.06.038>.
- [48] Ríos-Mercado RZ, Bard JF. The flow shop scheduling polyhedron with setup times. *J Comb Optim* 2003;7:291–318. <https://doi.org/10.1023/A:1027372722187>.
- [49] Dobson G. The cyclic lot scheduling problem with sequence-dependent setups. *Oper Res* 1992;40:736–49. <https://doi.org/10.1287/opre.40.4.736>.
- [50] Rajendran C, Ziegler H. Scheduling to minimize the sum of weighted flowtime and weighted tardiness of jobs in a flowshop with sequence-dependent setup times. *Eur J Oper Res* 2003;149:513–22. [https://doi.org/10.1016/S0377-2217\(02\)00485-X](https://doi.org/10.1016/S0377-2217(02)00485-X).
- [51] Radhakrishnan S, Ventura JA. Simulated annealing for parallel machine scheduling with earliness-tardiness penalties and sequence-dependent set-up times. *Int J Prod Res* 2000;38:2233–52. <https://doi.org/10.1080/00207540050028070>.
- [52] Haase K. Capacitated lot-sizing with sequence dependent setup costs. *Oper-Res-Spekt* 1996;18:51–9. <https://doi.org/10.1007/BF01539882>.
- [53] Kang S, Malik K, Thomas LJ. Lot sizing and scheduling on parallel machines with sequence-dependent setup costs. *Manage Sci* 1999;45:273–89. <https://doi.org/10.1287/mnsc.45.2.273>.
- [54] Meyr H. Simultaneous lot sizing and scheduling on parallel machines. *Eur J Oper Res* 2002;139:277–92. [https://doi.org/10.1016/S0377-2217\(01\)00373-3](https://doi.org/10.1016/S0377-2217(01)00373-3).
- [55] Clark AR. Optimization approximations for capacity constrained material requirements planning. *Int J Prod Econ* 2003;84:115–31. [https://doi.org/10.1016/S0925-5273\(02\)00400-0](https://doi.org/10.1016/S0925-5273(02)00400-0).
- [56] de Araujo SA, Arenales MN, Clark AR. Joint rolling-horizon scheduling of materials processing and lot-sizing with sequence-dependent setups. *J Heurist* 2007;13: 337–58. <https://doi.org/10.1007/s10732-007-9011-9>.
- [57] Ferreira D, Morabito R, Rangel S. Solution approaches for the soft drink integrated production lot sizing and scheduling problem. *Eur J Oper Res* 2009;196:697–706. <https://doi.org/10.1016/j.ejor.2008.03.035>.
- [58] Toso EAV, Morabito R, Clark AR. Lot sizing and sequencing optimisation at an animal-feed plant. *Comput Ind Eng* 2009;57:813–21. <https://doi.org/10.1016/j.cie.2009.02.011>.
- [59] Clark AR, Morabito R, Toso EAV. Production setup-sequencing and lot-sizing at an animal nutrition plant through atsp subtour elimination and patching. *J Schedul* 2010;13:111–21. <https://doi.org/10.1007/s10951-009-0135-7>.
- [60] Transchel S, Minner S, Kallrath J, Löndorf N, Eberhard U. A hybrid general lot-sizing and scheduling formulation for a production process with a two-stage product structure. *Int J Prod Res* 2011;49:2463–80. <https://doi.org/10.1080/00207543.2010.532910>.
- [61] Seeanner F, Meyr H. Multi-stage simultaneous lot-sizing and scheduling for flow line production. *OR Spect* 2013;35:33–73. <https://doi.org/10.1007/s00291-012-0296-1>.
- [62] Seeanner F, Almada-Lobo B, Meyr H. Combining the principles of variable neighborhood decomposition search and the fix&optimize heuristic to solve multi-level lot-sizing and scheduling problems. *Comput Oper Res* 2013;40:303–17. <https://doi.org/10.1016/j.cor.2012.07.002>.
- [63] Sioud A, Gagné C. Enhanced migrating birds optimization algorithm for the permutation flow shop problem with sequence dependent setup times. *Eur J Oper Res* 2018;264:66–73. <https://doi.org/10.1016/j.ejor.2017.06.027>.
- [64] Naderi B, Zandieh M, Shirazi MAHA. Modeling and scheduling a case of flexible flowshops: total weighted tardiness minimization. *Comput Ind Eng* 2009;57: 1258–67. <https://doi.org/10.1016/j.cie.2009.06.005>.
- [65] Li X, Yang Z, Ruiz R, Chen T, Sui S. An iterated greedy heuristic for no-wait flow shops with sequence dependent setup times, learning and forgetting effects. *Inf Sci (N Y)* 2018;453:408–25. <https://doi.org/10.1016/j.ins.2018.04.038>.
- [66] Jolai F, Rabiee M, Asefi H. A novel hybrid meta-heuristic algorithm for a no-wait flexible flow shop scheduling problem with sequence dependent setup times. *Int J Prod Res* 2012;50:7447–66. <https://doi.org/10.1080/00207543.2011.653012>.
- [67] Samarghandi H, ElMekkawy TY. Solving the no-wait flow-shop problem with sequence-dependent set-up times. *Int J Comput Integr Manuf* 2014;27:213–28. <https://doi.org/10.1080/0951192X.2013.812802>.
- [68] Nagano MS, da Silva AA, Nogueira Lorena LA. An evolutionary clustering search for the no-wait flow shop problem with sequence dependent setup times. *Expert Syst Appl* 2014;41:3628–33. <https://doi.org/10.1016/j.eswa.2013.12.013>.
- [69] Nagano MS, Miyata HH, Araújo DC. A constructive heuristic for total flowtime minimization in a no-wait flowshop with sequence-dependent setup times. *J Manuf Syst* 2015;36:224–30. <https://doi.org/10.1016/j.jmsys.2014.06.007>.
- [70] Gómez-Gasquet P, Andrés C, Lario F-C. An agent-based genetic algorithm for hybrid flowshops with sequence dependent setup times to minimise makespan. *Expert Syst Appl* 2012;39:8095–107. <https://doi.org/10.1016/j.eswa.2012.01.158>.
- [71] Vanchipura R, Sridharan R, Babu AS. Improvement of constructive heuristics using variable neighbourhood descent for scheduling a flow shop with sequence dependent setup time. *J Manuf Syst* 2014;33:65–75. <https://doi.org/10.1016/j.jmsys.2013.07.003>.
- [72] Meng L, Zhang C, Shao X, Zhang B, Ren Y, Lin W. More MILP models for hybrid flow shop scheduling problem and its extended problems. *Int J Prod Res* 2020;58: 3905–30. <https://doi.org/10.1080/00207543.2019.1636324>.
- [73] Li Y, Li X, Gao L, Zhang B, Pan Q-K, Tasgetiren MF, et al. A discrete artificial bee colony algorithm for distributed hybrid flowshop scheduling problem with sequence-dependent setup times. *Int J Prod Res* 2021;59:3880–99. <https://doi.org/10.1080/00207543.2020.1753897>.
- [74] Guo H, Sang H, Zhang B, Meng L, Liu L. An effective metaheuristic with a differential flight strategy for the distributed permutation flowshop scheduling problem with sequence-dependent setup times. *Knowl Based Syst* 2022;242: 108328. <https://doi.org/10.1016/j.knsys.2022.108328>.
- [75] Chen C-L. An iterated population-based metaheuristic for order acceptance and scheduling in unrelated parallel machines with several practical constraints. *Mathematics* 2023;11:1433. <https://doi.org/10.3390/math11061433>.
- [76] Mac Cawley A, Maturana S, Pascual R, Tortorella GL. Scheduling wine bottling operations with multiple lines and sequence-dependent set-up times: robust formulation and a decomposition solution approach. *Eur J Oper Res* 2022;303: 819–39. <https://doi.org/10.1016/j.ejor.2022.02.054>.
- [77] Alimian M, Ghezavati V, Tavakkoli-Moghaddam R, Ramezani R. Solving a parallel-line capacitated lot-sizing and scheduling problem with sequence-dependent setup time/cost and preventive maintenance by a rolling horizon method. *Comput Ind Eng* 2022;168:108041. <https://doi.org/10.1016/j.cie.2022.108041>.
- [78] Rohaninejad M, Hanzálek Z. Multi-level lot-sizing and job shop scheduling with lot-streaming: reformulation and solution approaches. *Int J Prod Econ* 2023;263: 108958. <https://doi.org/10.1016/j.ijpe.2023.108958>.
- [79] Lee Y, Lee K. Valid inequalities and extended formulations for lot-sizing and scheduling problem with sequence-dependent setups. *Eur J Oper Res* 2023;310: 201–16. <https://doi.org/10.1016/j.ejor.2023.03.003>.
- [80] de Freitas MG, Fuchigami HY. A new technology implementation via mathematical modeling for the sequence-dependent setup times of industrial problems. *Comput Ind Eng* 2022;172:108624. <https://doi.org/10.1016/j.cie.2022.108624>.

- [81] Costa A, Fernandez-Viagas V. A modified harmony search for the T-single machine scheduling problem with variable and flexible maintenance. *Expert Syst Appl* 2022;198:116897. <https://doi.org/10.1016/j.eswa.2022.116897>.
- [82] Feng Y, Hu Z, Si R, Wang J-B. Study on due-date assignment scheduling with setup times and general truncated learning effects. *Asia-Pac J Oper Res* 2024;41. <https://doi.org/10.1142/S0217595923500069>.
- [83] Song H, Liu P. A study on the optimal flexible job-shop scheduling with sequence-dependent setup time based on a hybrid algorithm of improved quantum cat swarm optimization. *Sustainability* 2022;14:9547. <https://doi.org/10.3390/su14159547>.
- [84] Park I-B, Huh J, Park J. A generation and repair approach to scheduling semiconductor packaging facilities using case-based reasoning. *IEEE Access* 2023;11:50631–41. <https://doi.org/10.1109/ACCESS.2023.3277529>.
- [85] Meng L, Zhang B, Gao K, Duan P. An MILP model for energy-conscious flexible job shop problem with transportation and sequence-dependent setup times. *Sustainability* 2022;15:776. <https://doi.org/10.3390/su15010776>.
- [86] Yan S, Zhang G, Sun J, Zhang W. An improved ant colony optimization for solving the flexible job shop scheduling problem with multiple time constraints. *Math Biosci Eng* 2023;20:7519–47. <https://doi.org/10.3934/mbe.2023325>.
- [87] Shen L. A tabu search algorithm for the job shop problem with sequence dependent setup times. *Comput Ind Eng* 2014;78:95–106. <https://doi.org/10.1016/j.cie.2014.09.004>.
- [88] Wang X, Dai Y, Wang L, Jia Z. Transient analysis and scheduling of Bernoulli serial lines with multi-type products and finite buffers. *IEEE Trans Automa Sci Eng* 2023;20:2367–82. <https://doi.org/10.1109/TASE.2022.3210259>.
- [89] Bruni ME, Khodaparasti S, Demeulemeester E. The distributionally robust machine scheduling problem with job selection and sequence-dependent setup times. *Comput Oper Res* 2020;123:105017. <https://doi.org/10.1016/j.cor.2020.105017>.
- [90] Bıçakcı PS, Derya T, Kara I. Solution approaches for the parallel machine order acceptance and scheduling problem with sequence-dependent setup times, release dates and deadlines. *Eur J Ind Eng* 2021;15:295. <https://doi.org/10.1504/EJIE.2021.115171>.
- [91] Tarhan İ, Oğuz C. A matheuristic for the generalized order acceptance and scheduling problem. *Eur J Oper Res* 2022;299:87–103. <https://doi.org/10.1016/j.ejor.2021.08.024>.
- [92] Mahmoudinazlou S, Alizadeh A, Noble J, Eslamdoust S. An improved hybrid ICA-SA metaheuristic for order acceptance and scheduling with time windows and sequence-dependent setup times. *Neural Comput Appl* 2024;36:599–617. <https://doi.org/10.1007/s00521-023-09030-w>.