

NO-WAIT FLOW SHOP SCHEDULING PROBLEM: A SYSTEMATIC LITERATURE REVIEW AND BIBLIOMETRIC ANALYSIS

DANA MARSETIYA UTAMA*, SABILA ZAHRA UMAMY AND CYNTHIA NOVEL AL-IMRON

Abstract. One of the most widely studied problems in flow shop scheduling is not allowing jobs to wait to be processed at the next stage. This constraint causes the job to be processed immediately at the next stage without waiting, so this problem is popularly called the No-Wait Flow Shop. This article aims to provide a comprehensive review of the No-Wait Flow Shop Scheduling (NWFS) problem based on a survey of published articles from 1999 to 2023. The article review is based on a systematic literature review, and bibliometric analysis is also presented based on the network processed using VOSviewer. One hundred twenty articles were collected from the Scopus database, which was reviewed based on NWFS variants, objective functions, and optimization procedures. The no-wait permutation flow shop scheduling (NWPFS) problem is a variant that researchers have widely investigated. Meta-heuristic procedures are widely applied to solve NWFS problems. In addition, the objective function of minimizing makespan is an objective function that researchers often apply. NWFS research gaps and future research trends are also presented in this paper.

Mathematics Subject Classification. 90-XX, 90-02.

Received October 30, 2023. Accepted December 26, 2023.

1. INTRODUCTION

In recent years, the increasing complexity and efficiency demands in the manufacturing industry have led to significant growth in research on the Flow Shop Scheduling Problem (FSP). This problem has become a significant focus of the scheduling literature, as indicated by Zaied *et al.* [1]. The FSP problem finds numerous applications in the manufacturing industry [2], particularly in manufacturing, automotive, electronics, and pharmaceuticals, where efficient scheduling is pivotal in managing production, directly impacting productivity and operational sustainability. Implementing an optimized solution for the FSP offers substantial benefits. We can determine the optimal sequence of jobs through particular scheduling based on the defined objective function [3]. Moreover, an efficient approach to FSP can significantly enhance the performance of manufacturing companies [4]. Therefore, gaining a profound understanding of FSP and developing suitable scheduling strategies are indispensable for achieving efficiency and competitive advantage in the modern manufacturing industry.

In today's manufacturing, the complexity of manufacturing processes poses a scheduling challenge [5, 6]. One of the primary scheduling constraints is the minimization of waiting time. Sometimes, the production process should progress without delays between production stages. It gives rise to the “No-Wait” scheduling,

Keywords. No-Wait, flow shop, scheduling, permutation, hybrid, distributed.

Department of Industrial Engineering, University of Muhammadiyah Malang, Malang, Indonesia.

*Corresponding author: dana@umm.ac.id

where no waiting time is allowed during job processing [7,8]. The “No-Wait” condition is significant in food or pharmaceutical production industries, where waiting time can impact product quality. Additionally, “No-Wait” flow shop scheduling finds applicability across diverse sectors, including the bakery industry, pharmaceutical processing, concrete goods production, and oil refineries. It entails continuous workflow through a machine, with no waiting time allowed between successive job operations. An illustration of this problem is depicted in Figure 1. This scheduling variant mandates that each job must proceed without waiting between machines. Once a job starts processing, it must be completed immediately until it reaches the last machine, with no waiting time between the job and the machine. The inability to optimize No-Wait Flow Shop Scheduling (NWFS) scheduling can result in increased process time, decreased productivity, higher production costs, and even customer dissatisfaction. Therefore, a deep understanding of overcoming the challenges in NWFS scheduling is essential for manufacturing companies seeking to improve their operational efficiency. This scheduling variant has garnered attention in concrete and metal production [9]. The primary challenge in this model is to create work sequences that adhere to the “No-Wait” criterion and optimize other performance indicators. Therefore, since Röck [10] introduced the production scheduling model with the No-Wait Flow Shop criterion, there has been significant progress in research and practical applications, underscoring the significance of this variant in various industrial sectors.

Flow shop problems with No-Wait constraints are called NWFS problems. The attention towards NWFS problems began in the 1980s, primarily focusing on optimizing classical objective functions. However, recent years have witnessed the evolution of this problem, leading to a diverse range of scheduling constraints and solution methods. With the emergence of new technologies, including artificial intelligence, the potential for enhancing NWFS solutions has become an increasingly attractive area of research. Recently, NWFS problems have garnered significant interest from researchers and practitioners. It is due to the inherent technical complexity of the problem and the substantial impact that improved scheduling efficiency can have on manufacturing operations. Numerous literature reviews on flow shop scheduling problems have been published. For instance, Rossit *et al.* [11] reviewed the Multi-objective Permutation Flow Shop Scheduling problem, while Yenisey and Yagmahan [12] discussed the non-permutation Flow Shop Scheduling problem. Komaki *et al.* [13] attempted to provide an overview of Flow Shop scheduling problems with assembly operations. Other comprehensive reviews encompass topics like the flow shop scheduling problem under Uncertainty [14], the blocking flow shop scheduling problem [15], and hybrid flow shop scheduling [16]. Recently, Neufeld *et al.* [17] presented a literature review on the Multi-objective Hybrid Flow Shop Scheduling problem, and Utama *et al.* [18] reviewed the energy efficiency of the hybrid flow shop scheduling problem. However, there is limited literature explicitly addressing the integration of current technologies into NWFS and how this can reshape scheduling practices in the manufacturing industry.

Several motivations underscore the need to review the NWFS problem: (1) NWFS problems are frequently applied in the manufacturing industry, and (2) there is no comprehensive review of the NWFS problem. (3) Furthermore, there is a pressing need to explore how recent technological advances can enhance the efficiency and effectiveness of addressing the NWFS problem. With these motivations in mind, this study aims to offer a comprehensive review of the NWFS problem. This literature review categorizes published NWFS articles based on NWFS variants, objective functions, and optimization procedures. Additionally, we conduct bibliometric analysis to map NWFS-related terms. This paper also presents recent trends in NWFS research. We identify and discuss areas requiring further research, guiding future researchers. Moreover, we highlight gaps in current research on unresolved NWFS issues and suggest future research directions within the NWFS domain. In summary, this paper substantially contributes to the field by delivering a comprehensive review of NWFS, drawing from published articles from 1999 to 2023.

The paper is structured as follows. Section 2 presents the systematic literature review method, which consists of planning, conducting, finding, and concluding stages. Section 3 presents a comprehensive review based on NWFS variants, objective function, and classification of optimization procedures. Bibliometric analysis is presented in Section 4. The analysis, gaps, and trends of NWFS research are described in Section 5. The paper ends with the conclusion and future research section.

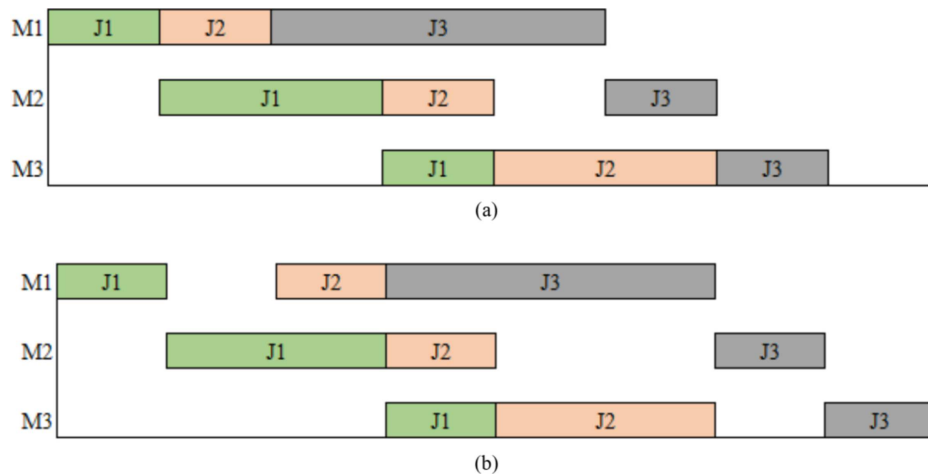


FIGURE 1. (a) Illustration of Permutation Flow shop problem; (b) illustration of No-Wait Permutation Flow shop problem.

TABLE 1. Research questions of the systematic review on NWFS.

ID	Research question	Aim
RQ1	What variants dominate NWFS research?	To identify the Variants of NWFS research.
RQ2	What procedures dominate NWFS research?	To identify the Procedures used in NWFS research.
RQ3	What objective functions do researchers often use for NWFS research?	To identify the Objective functions used by researchers for NWFS research.

2. METHODS

This article provides a review of NWFS issues that analyze previous research and can be used to guide future research. This review uses the systematic literature review (SLR) procedure. SLR reviews have three main stages: planning, conducting, and reporting [19]. Details of the Systematic literature review flow are presented in Figure 2. The description of the main stages for SLR of NWFS problems is presented as follows.

2.1. Planning stage

In this stage, the first part that must be done is the analysis of the need to conduct an SLR, and this is presented in the introduction section, which also reviews the urgency of SLR on NWFS issues. The second part is to determine research questions (RQ), which aims to present a review based on the RQ determined to focus on the research being investigated. The RQ used in this SLR can be seen in Table 1. The last stage is the review protocol. At this stage, a selection of databases is used to search for articles relevant to the topic of NWFS. The Scopus database was chosen in this SLR because it has a good reputation for indexing research. There were criteria set for the articles reviewed, namely full-text articles in English published in journals. The inclusion and exclusion criteria for SLR on the topic of NWFS can be seen in Table 2. Studies that proposed models and solution procedures for NWFS problems met the inclusion criteria. Then, studies that do not propose models and solution procedures for NWFS problems are identified as exclusion criteria.

TABLE 2. Inclusion and exclusion criteria on NWFS.

Inclusion criteria	Exclusion criteria
1. Primary Studies (Journal)	1. Article form Conferences and Secondary Studies
2. Contains peer-reviewed article	2. Systematic Review
3. Studies published between 1999 and 2023	3. Studies do not describe or propose a model algorithm, or procedure to solve NWPFS
4. Studies describing or proposing a model, algorithm, or procedure to solve NWFS?	4. Studies published before 1999
5. Full-text in English	–

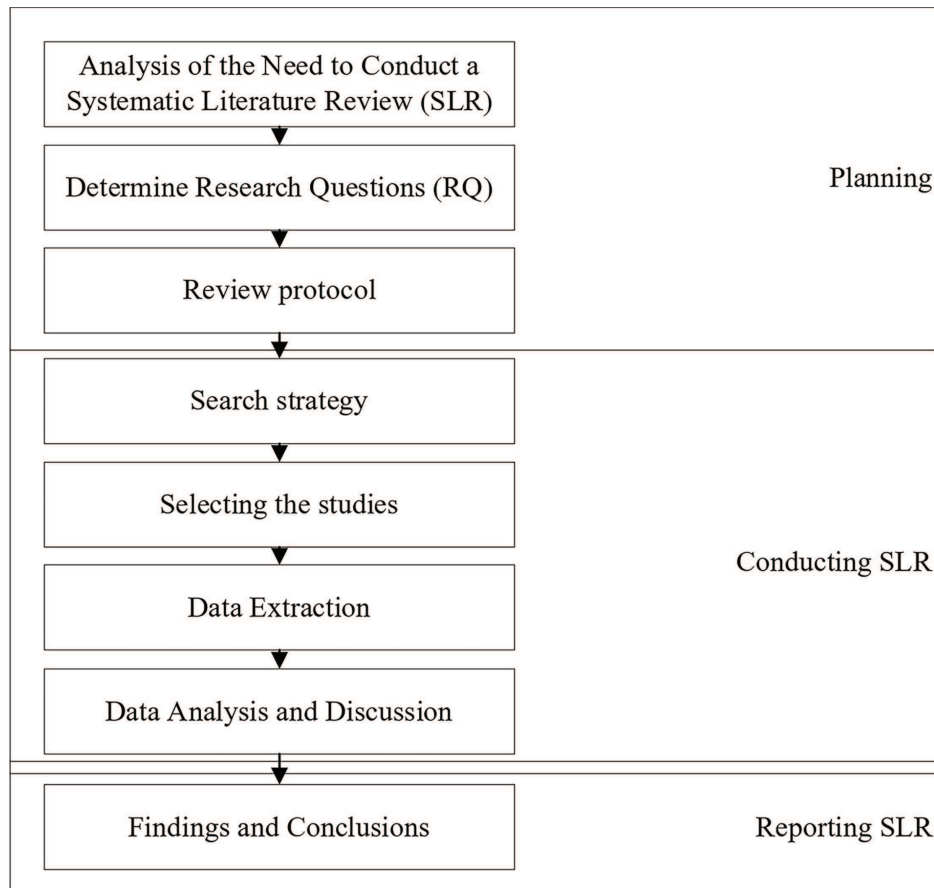


FIGURE 2. Systematic literature review stage.

2.2. Conducting SLR

In the Conducting SLR stage, several activities are carried out, such as determining the search strategy, selecting the studies, data extraction, and data analysis and discussion. The keywords are determined in the search for NWFS problem articles. The keyword determination method is based on Boolean procedures. The keywords used in this article search are “No”, AND “Wait”, AND “Flow”, “Shop”, “Scheduling”, AND “Prob-

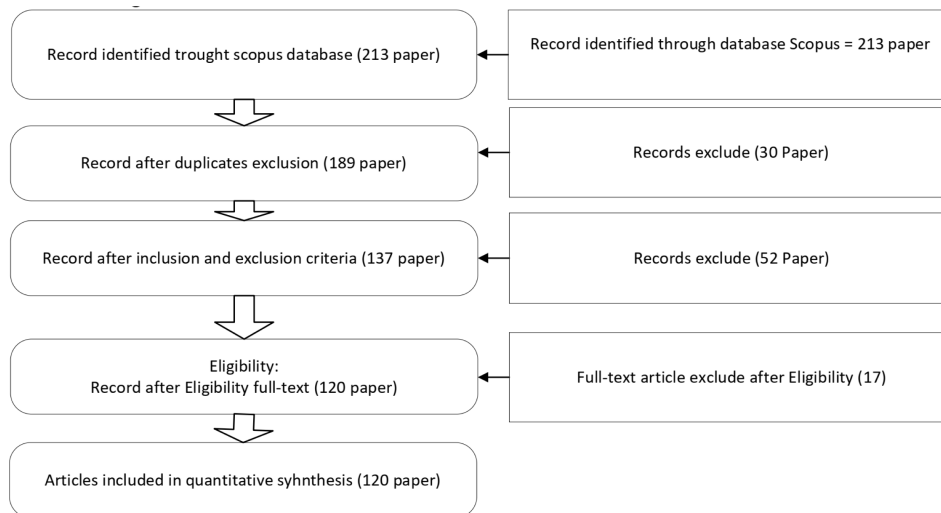


FIGURE 3. Screening article.

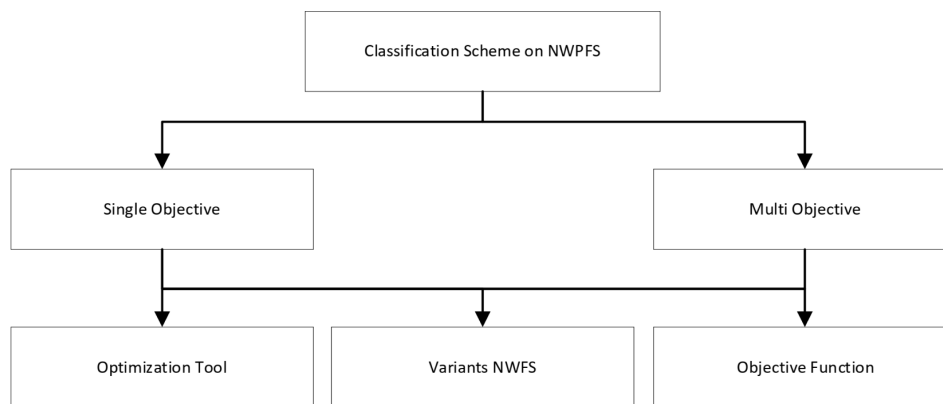


FIGURE 4. Literature Review classification scheme.

lem". Furthermore, study selection is carried out on the articles that have been collected. After the article search process using the Scopus database, 213 articles were obtained. However, some articles are duplicates, so the total number of articles obtained is 189. Then, 137 articles were obtained after applying inclusion and exclusion rules by reading the title and abstract of the article. The last stage is eligibility based on full-text articles and getting 120 relevant articles. The stages of this phase can be seen in Figure 3.

In the next stage, data extraction is based on 120 articles selected in the previous stage. Furthermore, analysis is carried out based on NWFS variants, optimization types (single-objective and multi-objective), optimization procedures, and objective function problems. The classification scheme of the literature review on NWFS can be seen in Figure 4. Bibliometric analysis is also presented and reviewed based on the year, journal, network, and overlay of words used in NWFS research. This analysis was processed using VOSviewer.

The SLR stage ended with data analysis and discussion. Data were analyzed using frequency distributions presented for each literature review classification scheme. This section also presents the research mapping based on the literature review classification scheme and the bibliometric analysis of the NWFS research.

2.3. Finding and conclusion stage

At this stage, the findings and conclusions are described. This stage is carried out to answer the research question classification that has been determined. In addition, this section presents research mapping based on the specified literature review classification scheme.

3. LITERATURE REVIEW

This section provides a comprehensive review of NWFS research based on the schema review described in Section 2.2. In the classification of optimization problems, these scheduling problems are classified into single-objective optimization and multi-objective optimization problems. Single objective NWFS optimization problems are NWFS problems where the optimal solution is based on a single objective function, such as minimization of makespan, tardiness, energy, and total flow time. In addition, multi-objective optimization problems are known as problems that solve several objective functions simultaneously, such as makespan and energy minimization or makespan and tardiness minimization [20].

Four classifications of procedure types are used to classify optimization procedures for NWFS problems: exact, heuristic, metaheuristic, and hybrid. Exact procedures are procedures for finding the best (optimal) solution to a problem that searches the entire space of possible solutions. As a result, exact procedures require much time for computation. Heuristics is a time-saving and efficient approach to solving problems but cannot guarantee that it produces the optimal solution [21]. Furthermore, metaheuristic procedures are sophisticated based on computational intelligence for complex optimization problems [22, 23]. Finally, Hybrid strategies are classified as procedures that combine several Heuristic, Metaheuristic, and Exact [24–26].

To classify NWFS variants, this review uses the taxonomy and nomenclature proposed by Graham *et al.* [27] to express NWFS scheduling variants. To describe the variant problem of the NWFS scheduling problem, a triplet notation $\alpha/\beta/\gamma$ is proposed. The first part α contains only one entry describing the machine environment or the flow shop production floor configuration. The β part describes the machine processing characteristics and shop floor constraints, allowing this part to have no entries, one or multiple entries. The last section (γ) describes the objective function. This section usually contains one or more entries. One entry indicates the problem is single-objective, while multiple entries indicate the multi-objective case. As discussed earlier, section α focuses on the configuration of NWFS variants. This paper successfully classifies seven NWFS variants as follows:

NWPFS : No-Wait Permutation Flow shop Scheduling
 NWHFS : No-Wait Hybrid Flow shop Scheduling
 DNWFS : Distributed No-Wait Flow Shop Scheduling
 MNWFS : Mixed No-Wait Flow Shop Scheduling
 ANWFS : Assembly No-Wait Flow Shop Scheduling
 DANWFS : Distributed Assembly No-Wait Flow Shop Scheduling
 DHNWFS : Distributed Heterogeneous No-Wait Flow Shop Scheduling

This paper presents formal definitions of key terminologies of scheduling problems based on these variants. In the scheduling context, a “flow shop” is a production system in which jobs pass through multiple process stages or machines in the same sequence. Each machine performs a different operation, and all jobs follow the same sequence from one machine to the next [28–30]. Meanwhile, in the context of a flow shop, “no-wait” refers to a situation where a job, once started, should continue without delay through all stages of the process or machines. There is no waiting time between successive operations on different machines [7]. The definition of “permutation” in flow shop scheduling refers to scheduling where the same sequence of jobs is maintained on all machines [31, 32]. If job *A* is processed before job *B* on the first machine, then *A* will always be processed before *B* on all other machines [33, 34].

Furthermore, a “hybrid flow shop” is a variation of a flow shop where some stages have more than one parallel machine [35]. In a hybrid flow shop, jobs pass through a series of stages, and at each stage, they can be processed

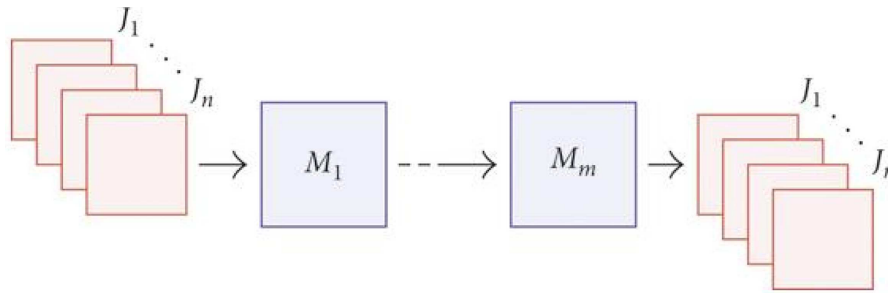


FIGURE 5. An illustration of NWPFS.

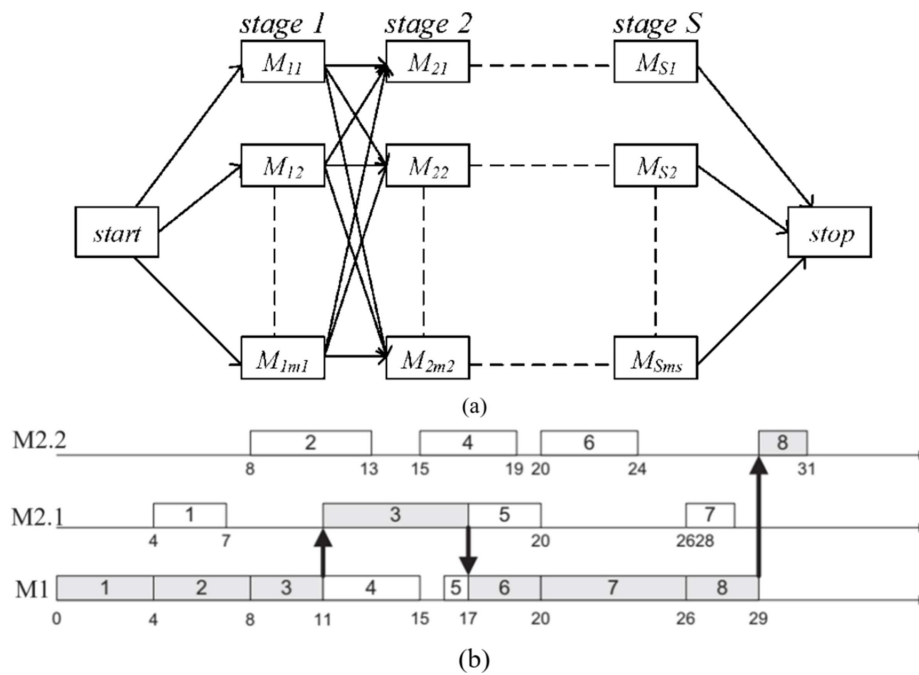


FIGURE 6. (a) An illustration of NWHFS; (b) illustration grant chart of the 2-stage NWHFS problem.

on one of several available machines [36]. Meanwhile, a “mixed no-wait flow shop” is a problem that combines elements of a traditional flow shop and a no-wait system. Some states may have no-wait requirements, while others do not [37]. It creates additional complexity in scheduling and coordinating jobs. “Assembly no-wait flow shop” is a production system where jobs involve assembling components or sub-units and must pass through various stages without delay [38]. It is usually applicable in industries where continuous assembly must be continuous to maintain quality or efficiency.

Meanwhile, a “distributed no-wait flow shop” is a flow shop setup where no-wait rules are applied, and production stages are spread across multiple locations or facilities [39]. The “distributed heterogeneous no-wait flow shop” problem, this problem combines the concepts of distributed flow shop and no-wait flow shop. Job must pass through a series of stages spread across multiple locations, with each location possibly having different characteristics or resources, and all of this must be done without waiting between stages [40]. Meanwhile, the

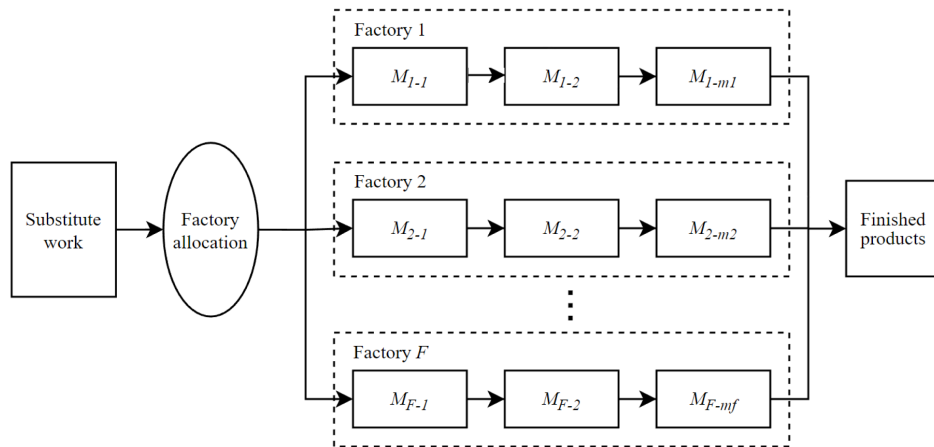


FIGURE 7. An illustration of Distributed No-Wait Flow Shop Scheduling (DNWFS).

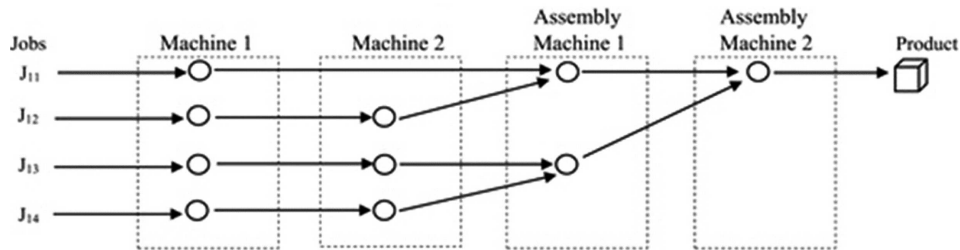


FIGURE 8. An illustration of the Assembly No-Wait Flow Shop Scheduling (ANWFS) problem.

“distributed assembly no-wait flow shop” is a variation of the assembly no-wait flow shop in which the assembly stages are spread across multiple locations. Jobs must pass through all these stages without waiting, even in different locations [41].

In the NWPFS problem, there are three variants, namely NWPFS, which is used to solve m machines; NWPFS-2 M, which is applied to 2 machines; and NWPFS-2/3M, which is used for 2/3 machines. The NWPFS problem is a classic NWFS job allocation problem where the job sequence for each machine is based on a permutation sequence [42]. An illustration of this problem is shown in Figure 5, with a Gantt chart illustration in Figure 1b. In this problem, each stage has a machine to process jobs with the same sequence of jobs on all machines. The NWHFS problem is a NWFS problem where it is an extension of NWPFS problem that allows each stage to have more than one machine [36]. Jobs move through several stages in this problem and can be processed on any of the machines available at each stage. The illustration of the NWHFS problem is shown in Figure 6a. Meanwhile, the Gantt chart illustration of the 2-stage NWHFS problem is shown in Figure 6b. On the MNWFS problem, this problem allows waiting time jobs on several stages and does not allow waiting time on some stages. This problem combines NWPFS and the classical permutation flow shop problem, where all jobs cannot wait on several machines [37]. The Gantt chart illustration of this problem combines Figures 1a and 1b.

Furthermore, the DNWFS problem is an extension of the NWFS problem, which considers the processing of NWFS at a single factory and allocating jobs to multiple factories [39]. An illustration of this problem is shown in Figure 7. In this problem, the main assumption is that in each factory, the machine specifications are the same (homogeneous). Furthermore, the DHNWF variant is a variant of the DNWFS evolution. The DHNWF variant is an extension of the DNWFS problem that considers allocating work to multiple factories with

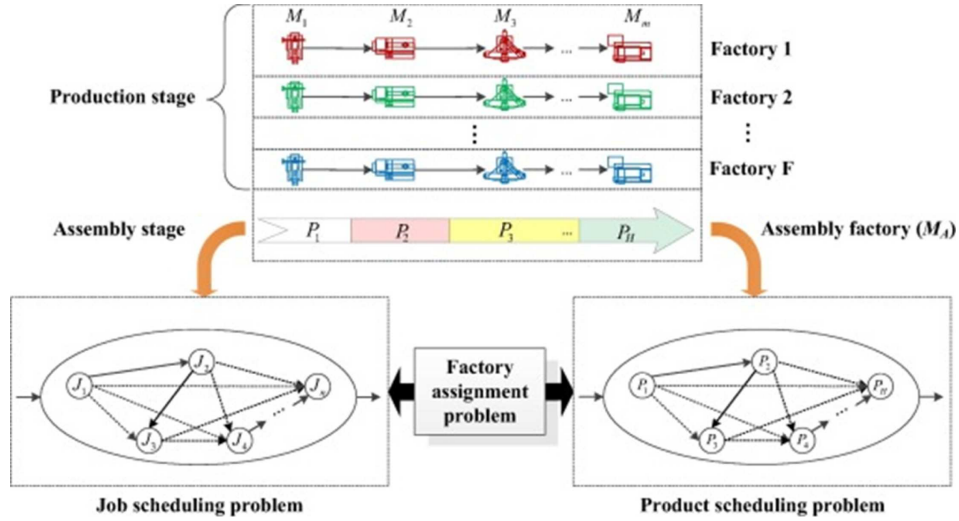


FIGURE 9. An illustration of Distributed Assembly No-Wait Flow Shop Scheduling (DANWFS).

heterogeneous machine specifications [40]. Therefore, this problem is increasingly complex due to its increasing complexity. The following variant is ANWFS. The ANWFS problem is an extension problem of NWPFS that also considers assembly processes [38]. This problem is illustrated in Figure 8. The last variant of NWFS is DANWFS. The DANWFS problem is an extension of the ANWFS problem that also decides the allocation of work to multiple factories [41]. This problem is illustrated in Figure 9.

The assumptions and constraints of the NWFS problem for some specific cases are discussed in subsection β . The fact that entries appear in subsection β indicates that the NWFS problem relies on relevant assumptions. Possible limitations and constraints include the following:

LMA : Limited Machine Availability
 MO : Missing Operations
 SMiES : Single machine in either stage
 NI : Nonavailability Interval
 DRD : Different Release Dates
 PR : Probable Rework
 BPM : Batch Processing Machines
 ST : Setup times
 TT : Transportation Times
 LE : Learning Effect
 SDST : Sequence-Dependent Setup Times
 UPM : Unrelated Parallel Machines
 NRUI : Non-Resumable Unavailable Interval
 FE : Forgetting Effects
 URD : Unequal Release Dates
 RT : Release Time
 DW : Due Windows
 UST : Uncertain Setup Times
 STV : Service time variation
 JR : Optional job rejection

The last symbol of the triplet notation (γ) describes the objective function of the NWFS problem. Here are some notations about the objective function:

Min Cmax : Minimize makespan
 Min T : Minimize tardiness
 Min TEC : Minimize total energy consumption
 Min TFT : Minimize total flowtime
 Min L : Minimize lateness
 Min E : Minimize earliness
 Min EC : Minimize energy cost
 Min DT : Minimize delay time
 Min AT : Minimize the accomplished time
 Min CT : Minimize cycle time
 Min WTV : Minimize waiting time variance

To illustrate the understanding of the notation used in NWFS, we describe the standard version of NWFS denoted as NWPFS/ α /Min Cmax. The notation indicates that in part α , the shop configuration and variants discussed by the researcher are NWPFS solved on m machines. Note that part β is empty. It indicates that there are no assumptions and constraints used in the problem. The last part, γ illustrates that the optimization criterion achieved is makespan minimization. Furthermore, a review of 120 published papers on NWFS Problems is presented in Table 3.

3.1. Single-objective problems NWFS

3.1.1. NWPFS problem

The NWPFS problem is the most investigated problem by researchers. We found how many Heuristic Procedures have been proposed to solve this problem. One variant of the NWPFS problem is to solve the two-machine problem. Espinouse *et al.* [43] solved this problem for two-machine with limited machine availability constraints. They proposed the Gilmore and Gomory algorithm with the objective function of minimizing the makespan. Glass *et al.* [42] also used the algorithm of Gilmore and Gomory by considering missing operations to minimize makespan. Still in the Two-machine problem, the algorithm of Gilmore and Gomory has been applied to minimize makespan with some constraints such as non-resumable unavailable interval proposed by Li *et al.* [44] and Unavailable Interval Constraints researched by Chen *et al.* [45]. Another heuristic procedure based on approximation algorithms was also developed by Espinouse *et al.* [46] with limited machine availability constraints to minimize makespan. The constraint of a non-availability interval was also considered by Kubzin and Strusevich [47] on the Two-machine NWPFS problem to minimize makespan. An approach based on a neuro-fuzzy inference system was developed by Shafaei *et al.* [48] to solve the two-machine NWPFS problem with the objective function of minimizing the makespan.

Various heuristic procedures have been proposed in the NWPFS problem for makespan minimization on m machines, such as approximation algorithms [49], fast composite heuristics [50], and Greedy algorithms [51]. Aldowaisan–Allahverdi algorithm [52], heuristic procedure based on short processing time [53], Nawaz–Enscore–Ham heuristic [54], and jigsaw puzzle-inspired heuristic [55] have also been proposed. With the same objective function, problems with sequence-dependent setup time constraints were also proposed by Almeida and Nagano [56]. They proposed an iterated greedy algorithm procedure to solve the NWPFS problem on m machines.

Three publications utilizing heuristic procedures have been proposed for NWPFS problems with the objective function of minimizing total flow time. Bertolissi [57] attempts to offer Job insertion algorithms, and Laha *et al.* [58] utilizes penalty-shift-insertion algorithms. Li *et al.* [59] proposed an iterated greedy algorithm procedure by considering sequence-dependent setup times, learning, and forgetting effects. The objective function they observed was to minimize total flow time. Furthermore, on the heuristic procedure, only Liu *et al.* [60] tried to minimize tardiness with the modified Nawaz–Enscore–Ham heuristic procedure.

TABLE 3. Classification of reviews on NWFS.

Year	Ref	Problems	Procedure clasification				Procedure optimization	Type	
			Heur	Meta	Hybrid	Exact		Single	Multi
1999	[43]	NWPFS-2M/LMA/Min Cmax	v	—	—	—	Algorithm of Gilmore and Gomory	v	—
1999	[42]	NWPFS-2M/—/Min Cmax	v	—	—	—	Algorithm of Gilmore and Gomory	v	—
2000	[57]	NWPFS/—/Min TFT	v	—	—	—	Job insertion algorithms	v	—
2001	[46]	NWPFS-2M/LMA/Min Cmax	v	—	—	—	Heuristic algorithms	v	—
2003	[36]	NWHFS/SMiES/Min Cmax	v	—	—	—	Greedy algorithm	v	—
2003	[49]	NWPFS/—/Min Cmax	v	—	—	—	Heuristic algorithm	v	—
2004	[47]	NWPFS-2M/NI/Min Cmax	v	—	—	—	Heuristic algorithm	v	—
2005	[71]	NWPFS/—/Min Cmax	—	v	—	—	Local search	v	—
2006	[68]	NWPFS-2/3M/—/Min Cmax	—	v	—	—	Simulated annealing or genetic algorithm	v	—
2007	[100]	NWPFS/—/Min Cmax	—	—	v	—	Integrated PSO, Annealing, NEH	v	—
2008	[131]	NWPFS/—/Min Cmax, Min T	—	v	—	—	Immune algorithm	—	v
2008	[132]	NWPFS/—/Min Cmax, Min T	—	v	—	—	Genetic algorithm	—	v
2008	[50]	NWPFS/—/Min Cmax	v	—	—	—	Heuristic algorithm	v	—
2008	[101]	NWPFS/—/Min Cmax	—	—	v	—	Hybrid Particle swarm optimization	v	—
2008	[61]	NWPFS-2M/—/Min Cmax	—	—	—	v	Branch and bound	v	—
2008	[135]	NWPFS/—/Min Cmax, Min TFT	—	v	—	—	Particle swarm optimization	—	v
2008	[51]	NWPFS/—/Min Cmax	v	—	—	—	Greedy algorithm	v	—
2008	[143]	NWPFS/—/Min Cmax, Min TFT	—	—	v	—	Hybrid evolutionary algorithm	—	v
2009	[139]	NWPFS/—/Min Cmax, Min T	—	—	v	—	Memetic algorithm based on differential evolution	—	v
2009	[154]	NWPFS/—/Min Cmax, Min L	—	—	v	—	Genetic algorithms and iterated greedy procedures	—	v
2009	[52]	NWPFS/—/Min Cmax	v	—	—	—	Heuristic algorithm	v	—
2009	[141]	NWPFS/—/Min Cmax, Min T	—	—	v	—	Differential evolution algorithm and neighborhood	—	v
2009	[102]	NWPFS/—/Min Cmax	—	—	v	—	Differential evolution algorithm and neighborhood	v	—
2011	[48]	NWPFS-2M/—/Min Cmax	v	—	—	—	Neuro fuzzy inference system	v	—
2010	[111]	NWPFS/—/Min L	—	—	v	—	Tabu search and neighborhood	v	—
2010	[97]	NWPFS/—/Min L	—	v	—	—	Tabu search	v	—
2011	[62]	NWPFS-2M/NI, DRD/Min Cmax	—	—	—	v	Branch-and-bound	v	—
2011	[92]	NWPFS/—/Min TFT	—	v	—	—	Harmony search algorithm	v	—
2011	[142]	NWPFS/—/Min Cmax, Min TFT	—	—	v	—	Variable Neighborhood Search, genetic algorithm	—	v
2012	[106]	NWPFS/—/Min Cmax	—	—	v	—	Tabu Search and Particle Swarm Optimisation	v	—
2012	[108]	NWPFS/SDST, PR/Min Cmax	—	—	v	—	Genetic algorithm and simulated annealing	v	—
2012	[73]	NWPFS/—/Min Cmax	—	v	—	—	Genetic algorithm	v	—
2012	[69]	NWPFS-2M/BPM/Min Cmax	—	v	—	—	Particle Swarm Optimisation	v	—
2012	[112]	NWPFS/—/Min T	—	—	v	—	Simulated annealing and genetic algorithms	v	—
2012	[117]	NWPFS/ST/Min TFT	—	—	v	—	Genetic Algorithm–Cluster Search	v	—
2012	[113]	NWPFS/—/Min T	—	—	v	—	Hybrid Heuristic	v	—
2012	[114]	NWPFS/—/Min TFT	—	—	v	—	Hybrid particle swarm optimization algorithm	v	—
2012	[133]	NWPFS//Min Cmax, Min T	—	v	—	—	Electromagnetism algorithm	—	v
2012	[109]	NWPFS/SDST/Min Cmax	—	—	v	—	Genetic algorithm, Simulated annealing and Greedy algorithm	v	—
2013	[53]	NWPFS/—/Min Cmax	v	—	—	—	Heuristic Algorithm	v	—
2013	[60]	NWPFS/—/Min T	v	—	—	—	Nawaz–Encscore–Ham heuristic	v	—
2013	[38]	ANWFS/—/Min Cmax	—	—	v	—	Genetic, Differential Evolution Algorithm variable neighborhood search Algorithm	v	—
2013	[70]	NWPFS-2M/ST/Min L	—	v	—	—	Genetic algorithm	v	—
2015	[118]	NWHFS/—/Min Cmax	—	—	—	v	Branch and bound	v	—
2014	[87]	NWPFS/SDST/Min Cmax	—	v	—	—	Evolutionary clustering search	v	—
2014	[122]	NWHFS/—/Min Cmax	—	v	—	—	Hunting search algorithm	v	—
2014	[88]	NWPFS/SDST/Min Cmax	—	v	—	—	Particle Swarm Optimisation	v	—
2014	[115]	NWPFS/—/Min TFT	—	—	v	—	particle swarm optimization algorithm and memetic algorithm	v	—
2014	[58]	NWPFS/—/Min TFT	v	—	—	—	Heuristic algorithm	v	—
2015	[110]	NWPFS/SDST, LE/Min Cmax	—	—	v	—	Variable neighborhood search, Tabu Search Simulated Annealing	v	—
2015	[103]	NWPFS/—/Min Cmax	—	—	v	—	Iterated greedy algorithm and Tabu search	v	—
2015	[74]	NWPFS/—/Min Cmax	—	v	—	—	Genetic algorithm	v	—
2015	[75]	NWPFS/—/Min Cmax	—	v	—	—	Particle Swarm Optimisation	v	—
2015	[89]	NWPFS/SDST/Min Cmax	—	v	—	—	Genetic algorithm	v	—

TABLE 3. continued.

Year	Ref	Problems	Procedure clasification				Procedure optimization	Type	
			Heur	Meta	Hybrid	Exact		Single	Multi
2016	[54]	NWPFS/-/Min Cmax	v	-	-	-	Nawaz-Enscore-Ham heuristic	v	-
2016	[83]	NWPFS/-/Min Cmax	-	v	-	-	Teaching-learning based optimization algorithm	v	-
2016	[123]	NWHFS/UPM/Min Cmax	-	v	-	-	Biogeography-based optimisation algorithm	v	-
2016	[93]	NWPFS/-/Min TFT	-	v	-	-	Fast local neighbourhood search algorithm	v	-
2016	[127]	NWPFS/-/Min Cmax, Min TFT	v	-	-	-	Penalty-based construction algorithm	-	v
2016	[151]	NWHFS/-/Min Cmax, Min DT	-	v	-	-	Particle swarm optimization	-	v
2017	[94]	NWPFS/-/Min TFT	-	v	-	-	Simulated annealing	v	-
2017	[125]	DNWFS/-/Min Cmax	-	-	v	-	Iterated greedy algorithm and Neighborhood structures Local search	v	-
2017	[78]	NWPFS/-/Min Cmax	-	v	-	-	Learning phase Local search	v	-
2017	[39]	DNWFS/-/Min Cmax	-	v	-	-	Variable Neighbourhood Search algorithm	v	-
2017	[65]	NWPFS/-/Min Cmax	-	-	-	v	Branch and bound	v	-
2017	[44]	NWPFS-2M/NRUI/Min Cmax	v	-	-	-	Algorithm of Gilmore and Gomory	v	-
2017	[76]	NWPFS/-/Min Cmax	-	v	-	-	Particle Swarm Optimization	v	-
2019	[79]	NWPFS/-/Min Cmax	-	v	-	-	Water wave optimization	v	-
2018	[104]	NWPFS/-/Min Cmax	-	-	v	-	Hybrid ant colony algorithm	v	-
2018	[140]	NWPFS/-/Min Cmax, Min T	-	-	v	-	Simulated annealing and insertion algorithm	-	v
2018	[59]	NWPFS/SDST, LE, FE/Min TFT	v	-	-	-	Greedy algorithm	v	-
2017	[37]	MNWFS/-/Min Cmax	v	-	-	-	Greedy algorithm	v	-
2018	[136]	NWPFS/-/Min Cmax, Min TFT	-	v	-	-	Ant colony algorithm	-	v
2018	[63]	NWPFS-2M/URD, NI/Min Cmax	-	-	-	v	Branch-and-bound	v	-
2018	[91]	NWPFS/-/Min TFT	-	v	-	-	Genetic algorithm	v	-
2018	[95]	NWPFS/-/Min AT	-	v	-	-	Flower pollination algorithm	v	-
2018	[105]	NWPFS/-/Min Cmax	-	-	v	-	Water Wave Optimization and greedy algorithm	v	-
2018	[120]	NWHFS/-/Min Cmax	v	-	-	-	Heuristic algorithm	v	-
2019	[107]	NWPFS/-/Min Cmax	v	-	-	-	Hybrid biogeography-based optimization and variable neighborhood search	v	-
2019	[146]	DNWFS/SDST/Min Cmax, Min T	v	-	-	-	Pareto-based estimation of distribution algorithm	-	v
2019	[77]	NWPFS/-/Min Cmax	-	v	-	-	Particle swarm optimization	v	-
2019	[80]	NWPFS/-/Min Cmax	-	v	-	-	Cuckoo search algorithm	v	-
2019	[99]	NWPFS/LE/Min TEC	-	v	-	-	League Champions algorithms	v	-
2019	[72]	NWPFS/-/Min Cmax	-	v	-	-	Variable neighborhood search	v	-
2020	[128]	NWPFS/-/Min Cmax, Min T	-	-	-	v	Brand and Bound	-	v
2020	[45]	NWPFS-2M/NI/Min Cmax	v	-	-	-	Algorithm of Gilmore and Gomory	v	-
2020	[155]	NWPFS/-/Min Cmax, Min TEC	v	-	-	-	Variable neighborhood search	-	v
2020	[129]	NWPFS/-/Min T, Min E	-	-	-	v	Branch and Bound	-	v
2020	[96]	NWPFS/SDST, RT/Min T	-	v	-	-	Differential evolution	v	-
2020	[81]	NWPFS/-/Min Cmax	-	v	-	-	Discrete migrating birds optimization	v	-
2020	[55]	NWPFS/-/Min Cmax	v	-	-	-	Heuristic algorithm	v	-
2021	[126]	DANWFS/-/Min TFT	v	-	-	-	Backtracking search algorithm	v	-
2021	[82]	NWPFS/-/Min Cmax	-	v	-	-	Cuckoo search algorithm	v	-
2021	[144]	NWPFS/-/Min T, Min TFT	-	-	v	-	Genetic algorithm and global search algorithm	-	v
2021	[40]	DHNWFS/-/Min Cmax	-	v	-	-	Artificial Bee Colony	v	-
2021	[121]	NWHFS/-/Min Cmax	v	-	-	-	Heuristic algorithm	v	-
2021	[90]	NWPFS/-/Min Cmax	-	v	-	-	Simulated Annealing Algorithm	v	-
2021	[124]	NWHFS/UPM/Min TFT	-	-	v	-	Mixed integer linear programming model dan genetic simulated annealing algorithm	v	-
2021	[145]	NWPFS/-/Min T, Min EC	-	-	v	-	Mixed integer linear programming model dan Multi-objective Variable Neighborhood	-	v
2022	[148]	DNWFS/DW/Min L, Min E	-	v	-	-	A discrete fruit fly optimization algorithm	-	v
2022	[98]	NWPFS/MO/Min CT	-	v	-	-	Tabu search	v	-
2022	[147]	DNWFS/SDST/Min Cmax, Min T	-	v	-	-	Genetic algorithm, artificial bee colony algorithm, migratory bird optimization	-	v
2022	[116]	NWPFS/-/Min TFT	-	-	v	-	Greedy algorithm, simulated annealing, and variable neighbourhood search	v	-
2022	[134]	NWPFS/SDST/Min T, Min E	-	v	-	-	Genetic algorithm	-	v
2022	[67]	NWPFS-2M/-/Min Cmax	-	v	-	-	Tabu search	v	-
2022	[119]	NWHFS/-/Min T	-	-	-	v	Mixed integer linear programming model	v	-
2022	[64]	NWPFS-2M/UST/Min Cmax	-	-	-	v	Branch and Bound	v	-

Six studies with exact procedures have been presented on NWPFS problems with various constraints. All the proposed procedures are based on the Branch and bound approach with the objective function of minimizing the makespan. Su and Lee [61] investigated a two-machine NWPFS problem involving a single server. With

TABLE 3. continued.

Year	Ref	Problems	Procedure classification				Procedure optimization	Type	
			Heur	Meta	Hybrid	Exact		Single	Multi
2023	[56]	NWPFS/SDST/Min Cmax	v	–	–	–	Greedy algorithm	v	–
2023	[41]	DANWFS/–/Min TFT	v	–	–	–	Greedy algorithm	v	–
2023	[149]	DNWFS/SDST/Min Cmax, Min TEC	–	–	v	–	Hybrid meta-heuristic algorithm based on Q-learning	–	v
2023	[150]	DNWFS/SDST/Min Cmax, Min TEC	–	–	v	–	Non-dominated sorting genetic algorithm and Nawaz-Enscore-Ham heuristic	–	v
2023	[137]	NWPFS/SDST, RT/Min T, Min L	–	v	–	–	Matrix-Cube-Based Estimation of Distribution Algorithm	–	v
2023	[152]	DANWFS/–/Min Cmax Min TEC	–	v	–	–	A reinforcement learning-driven brain storm optimisation	–	v
2023	[66]	NWPFS-2S /STV-JR/Min WTV	–	–	–	v	Dynamic programming	v	–
2023	[153]	DANWFS/–/Min Cmax, Min TEC	v	–	–	–	Iterative greedy	–	v
2023	[84]	NWPFS/–/Min Cmax	–	v	–	–	Whale optimization algorithm	v	–
2023	[138]	NWPFS/–/Min T, Min E	–	v	–	–	Simulated Annealing	–	v
2023	[85]	NWPFS/–/Min Cmax	–	v	–	–	Teaching-Learning-Based Optimization Algorithm	v	–
2023	[86]	NWPFS/–/Min Cmax	–	v	–	–	Discrete state transition algorithm	v	–
2023	[130]	NWPFS/SDST/Min Cmax, Min T	v	–	–	–	Iterative greedy	–	v

constraints under non-availability constraints and different release dates, Ben Chihoui *et al.* [62] investigated the two-machine NWPFS problem. The two-machine NWPFS problem was also investigated by Labidi *et al.* [63] by considering unequal release dates and non-availability constraints. The involvement of uncertain setup time constraints was also investigated by Allahverdi [64] to solve the two-machine NWPFS problem. Furthermore, only Samarghandi and Behroozi [65] investigated NWPFS on machines using branch and bound. Recently, Koulamas and Kyparisis [66] offered a dynamic programming procedure in solving the NWPFS two-stage problem considering minimal service time variation and optional job rejection. The problem focuses on the objective of minimizing waiting time variance.

In metaheuristic procedures, various procedures have been proposed to solve the NWPFS problem. In the two/three machine problem, Azerine *et al.* [67] proposed a Tabu search procedure, and Wang *et al.* [68] developed Simulated annealing and genetic algorithm to minimize the makespan. Some constraints on NWPFS problems are also presented. Muthuswamy *et al.* [69] involved batch processing machines to minimize makespan with particle swarm optimization procedure. Pang [70] also proposes a genetic algorithm to minimize lateness. In the NWPFS m machine problem, various algorithms are proposed to minimize the makespan, such as Local search [71], Variable neighborhood search [72], Genetic algorithm [73, 74], and particle swarm optimization [75–77]. Learning phase Local search [78], single water wave optimization [79], cuckoo co-search algorithm [80], An improved discrete migrating birds optimization [81], cuckoo co-evolutionary algorithm [82], and teaching-learning based optimization [83] have also been proposed to minimize makespan. Recently, advanced metaheuristic algorithms such as the Whale optimization algorithm [84], Teaching-Learning-Based Optimization Algorithm [85], and Discrete state transition algorithm [86] have been proposed to solve the makespan minimization problem in NWPFS. Researchers have also proposed NWPFS m machine problems involving various constraints. Nagano *et al.* [87] proposed the sequence-dependent setup times constraint to minimize makespan with an Evolutionary clustering search procedure. Samarghandi and ElMekkawy [88] also investigated the same constraint with the objective function of minimizing makespan with particle swarm optimization procedure. Samarghandi [89] also proposes a genetic algorithm to minimize makespan by involving sequence-dependent setup times. Sun *et al.* [90] investigated constraints considering Interval-Valued Fuzzy Processing Time to minimize makespan with the Annealing Algorithm procedure.

The objective function to minimize total flowtime has also been investigated with various metaheuristic procedures on the NWPFS m machine problem. The proposed procedures include a Genetic algorithm [91], harmony search algorithm [92], fast local neighborhood search algorithm [93], and simulated annealing [94]. Furthermore, the flower pollination algorithm [95] was also proposed to minimize accomplished time. Minimized tardiness was investigated by Hu *et al.* [96] on the NWPFS problem by considering sequence-dependent setup

times and release times. They proposed an enhanced differential evolution procedure. Qian [97] utilized Tabu search to minimize latency, and Smutnicki *et al.* [98] also used Tabu search with missing operations constraints to minimize cycle time. Lastly, League Champions algorithms were proposed by Nouri *et al.* [99] to solve NWPFS with Learning effects constraints. Their study focuses on minimizing total energy consumption.

Hybrid procedures are also popular procedures for solving NWPFS problems. Several procedures are applied to minimize the makespan, such as Integrated Particle Swarm-simulated Annealing–Nawaz–Enscore–Ham heuristic [100], Hybrid Particle swarm optimization [101], Differential Evolution-Neighborhood [102], Iterated greedy algorithm-Tabu search [103]. Prosedur based on a hybrid ant colony algorithm [104], Water Wave Optimization-Iterated greedy algorithm [105], Tabu Search-Particle Swarm Optimisation [106], and hybrid biogeography-based optimization with variable neighborhood search [107] were also offered to minimize makespan. Sequence-dependent setup times and probable rework constraints were developed by Rabiee *et al.* [108] based on an integrated genetic algorithm and simulated annealing. The objective function that their research investigated was makespan minimization. With the same objective function, Ying *et al.* [109] developed a combination of Genetic algorithm, Simulated annealing, and Greedy algorithm with sequence-dependent setup times constraints. The same constraint by adding position-based learning consideration was also investigated by Jabbari and Azizi [110] using Variable neighborhood search, Tabu Search Simulated Annealing with the objective function of minimizing the makespan. Other objective functions based on the due date concept have also been proposed to minimize latency and tardiness. Tabu search and neighborhood integration was developed to minimize lateness [111]. Furthermore, the integration of Simulated annealing and genetic algorithms [112] and Hybrid Heuristic [113] has been used to minimize tardiness. Several researchers also investigated the objective function of minimizing total flowtime. Several procedures have been proposed, such as the Hybrid particle swarm optimization algorithm [114], particle swarm optimization algorithm and memetic algorithm [115], and Greedy algorithm-simulated annealing-variable neighborhood search [116]. Minimizing total flow time by considering setup times was also investigated by Nagano *et al.* [117] by utilizing Genetic Algorithm-Cluster Search.

3.1.2. NWHFS problem

In the NWHFS problem, two exact procedures were proposed to solve this problem. In the two-stage NWHFS problem, Wang *et al.* [118] proposed a Branch and bound procedure to minimize makespan. With the objective function of minimizing tardiness, Azaiez *et al.* [119] solved the two-stage NWHFS problem with inter-stage flexibility constraints with a mixed integer linear programming procedure. Furthermore, a Heuristic procedure is proposed to minimize the makespan. Liu *et al.* [36] solved the two-stage NWHFS problem with the constraint of a single machine in either stage with a Greedy algorithm procedure. A heuristic procedure was also proposed based on a polynomial time approximation algorithm involving inter-stage flexibility in the NWHFS problem [120]. Dong *et al.* [121] also proposed a heuristic algorithm to solve the two-stage NWHFS problem with the constraint of the flexibility of the first machine to minimize makespan. Other procedures based on metaheuristics and hybrid are also proposed to solve the NWHFS problem. Naderi *et al.* [122] proposed a hunting search algorithm to minimize the makespan. Rabiee *et al.* [123] developed a metaheuristic biogeography-based optimization procedure to minimize the makespan. Their research focused on NWHFS with the constraint of unrelated parallel machines. Xuan *et al.* [124] also investigated the same constraint by proposing a Mixed integer linear programming model and genetic-simulated annealing algorithm. Their research tries to minimize the total flow time.

3.1.3. DNWFS and DANWFS problem

This section presents the DNWFS and DANWFS issues. We note 2 studies each that address these problems. Shao *et al.* [125] investigated the DNWFS problem to minimize makespan. They developed a hybrid procedure based on an Iterated greedy algorithm-Neighborhood structures-Local search procedure. A variable Neighborhood Search algorithm based on metaheuristics was proposed by Komaki and Malakooti [39] to solve the DNWFS problem with the objective function of minimizing the makespan. In the DANWFS problem, Zhao *et al.* [126] proposed a Backtracking search algorithm to minimize total flowtime. With the same objective

function, Zhao *et al.* [41] proposed a population-based iterated greedy algorithm procedure for the DANWFS problem.

3.1.4. Other variant problem

Some other variations of the NWFS problem are also presented in this section. Mozdgir *et al.* [38] investigated the two-stage ANWFS problem with a makespan minimization objective function. They proposed a hybrid procedure based on Genetic Algorithm-Differential Evolution-variable neighborhood search. In the MNWFS problem, Wang *et al.* [37] utilized an iterated greedy algorithm procedure to minimize makespan. Finally, the problem is investigated by DHNWFS, which focuses on minimizing makespan with an Artificial Bee Colony procedure.

3.2. Multi-objective problems NWFS

3.2.1. NWPFS problem

This section describes the multi-objective research on the NWPFS problem. This problem is the most researched. A heuristic procedure based on a penalty-based construction algorithm is proposed by Laha and Gupta [127] to minimize makespan and total flowtime. Furthermore, two exact procedures based on Branch and bound are also proposed to solve the NWPFS problem. The objective functions investigated are minimizing makespan and tardiness by Madhushini and Rajendran [128] and reducing tardiness and earliness by Schaller and Valente [129]. Most recently, de Almeida and Nagano [130] proposed an iterative greedy procedure to solve the NWPFS problem that considers sequence-dependent setup times to minimize tardiness and makespan.

Metaheuristic procedures are widely used to solve NWPFS problems. Several algorithms have been developed to minimize makespan and tardiness, such as the Immune algorithm [131] and the Genetic algorithm [132]. Khalili [133] also investigated minimizing makespan and tardiness on NWPFS problems involving transportation times. They developed a metaheuristic procedure based on the principle of the electromagnetism algorithm. The sequence-dependent setup times constraint of the NWPFS problem was investigated by Guevara-Guevara *et al.* [134] by proposing a Genetic algorithm to minimize tardiness and earliness. Particle swarm optimization [135] and ant colony optimization algorithm [136] were proposed to solve the NWPFS problem in the research focusing on minimizing makespan and total flow time. The Matrix-Cube-Based Estimation of the Distribution Algorithm procedure was proposed by Qian *et al.* [137] on NWPFS problems that consider Sequence-Dependent Setup Times and Release Times. Their research focuses on minimizing tardiness and lateness. Karacan *et al.* [138] recently offered a Simulated Annealing procedure to solve the NWPFS problem to minimize tardiness and earliness.

Various hybrid procedures have also been proposed to solve NWPFS problems. Research that focuses on minimizing makespan and minimizing tardiness has proposed various procedures such as memetic algorithm (MA) based on differential evolution [139], simulated annealing and insertion algorithm [140], and differential evolution algorithm-neighborhood [141]. A variable Neighborhood Search algorithm, genetic algorithm [142], and hybrid evolutionary algorithm [143] were also developed to minimize makespan and total flowtime. Keskin and Engin also used the genetic algorithm and global search algorithm [144] to minimize tardiness and total flow time. Finally, Gomes *et al.* [145] investigated minimizing tardiness and energy cost by developing a Mixed integer linear programming model and Multi-objective Variable Neighborhood Search procedures.

3.2.2. DNWFS problem

This section describes the multi-objective problem in DNWFS. Five publications are recorded addressing this problem. Shao *et al.* [146] proposed a Pareto-based estimation of the distribution algorithm to minimize makespan and tardiness. Their research considered sequence-dependent setup time in DNWFS. With the same constraints, Allali *et al.* [147] developed a hybrid procedure based on a Genetic algorithm, artificial bee colony algorithm, and migratory bird optimization algorithm to minimize makespan and tardiness. DNWFS with due windows was investigated by Zhu *et al.* [148] to minimize latency and earliness. A discrete knowledge-guided

learning fruit fly optimization algorithm was proposed to solve this problem. The problem with sequence-dependent setup time constraints to minimize makespan minimize total energy consumption was also observed. The hybrid meta-heuristic algorithm based on Q-learning [149], the Non-dominated sorting genetic algorithm, and the Nawaz–Enscore–Ham heuristic [150] are developed to solve this problem.

3.2.3. Other problem

Other variants of multi-objective problems are NWHFS and DANWFS. Qin and Zhang [151] solved the NWHFS problem by minimizing makespan and delay time. They developed an elite particle swarm optimization procedure to solve this problem. In the DANWFS problem, Zhao *et al.* [152] proposed A reinforcement learning-driven brainstorm optimization to minimize makespan and total energy consumption. Recently, Zhao *et al.* [153] proposed an iterative greedy procedure to solve the DANWFS problem with the objective functions of minimizing makespan and total energy consumption.

4. BIBLIOMETRIC ANALYSIS

4.1. Paper distribution based on year

This section analyzes the articles' development based on the publication year. The distribution of NWFS articles by year can be seen in Figure 10. Based on bar graph data, a bibliometric analysis of NWFS research from 1999 to 2023 reveals essential insights into the evolution and interest in this field. There is an upward trend in annual publications, signaling a continuous increase in interest in NWFS in both academic and industrial environments. Initially, from 1999 to 2009, the limited number of publications reflects that NWFS may be a new area that has yet to be widely explored. It could be due to limitations in resources or technology at that time. However, there was a significant spike in 2010 with nine publications, marking an important turning point in NWFS research, which may be related to technological advances or discoveries that meet industry needs.

Interestingly, there were fluctuations in the number of publications each year, indicating changes in research priorities or funding allocations. Notably, there was a significant increase in publications between 2020 and 2023, with 2023 recording the highest number of publications. It could indicate renewed interest, possible breakthroughs in research, or increased opportunities for publication in conferences and journals. The five-year interval analysis also shows a consistent increase in publications, confirming the growth of interest and development in this field. These changes in the number of publications can be attributed to external factors such as technological developments and industry needs.

4.2. Distribution of articles by journal

This section presents the distribution of articles by the journal, as shown in Table 4. The results of the quantitative analysis show that the International Journal of Advanced Manufacturing Technology contributed the most articles, with a total of 11 articles. Then, the International Journal of Production Research is ranked second with a total distribution of 9 articles. Furthermore, swarm and Computers and Operations Research journals ranked third and fourth with six articles each.

4.3. Keywords and term analysis

This section presents the term and keyword analysis presented in a network. This network analysis was used to identify essential words related to NWFS research and analyze the relationship between words. This network analysis is processed using VOSviewer software, which is generated by performing a co-occurrence analysis of keywords. The advantage of this analysis is that it helps researchers reveal the development and structure of the NFWF research field to assist in mapping the intellectual structure of the NWFS scientific literature. The keywords analyzed were obtained from the titles and abstracts of the publications.

The visualized cluster distribution of the co-words network on NWFS is depicted in Figure 11. The results show that some essential keywords frequently used in papers are No-wait Flow shop Scheduling, Flow shop Scheduling, and Makespan. The larger the circle of the word indicates that the words are frequently used in

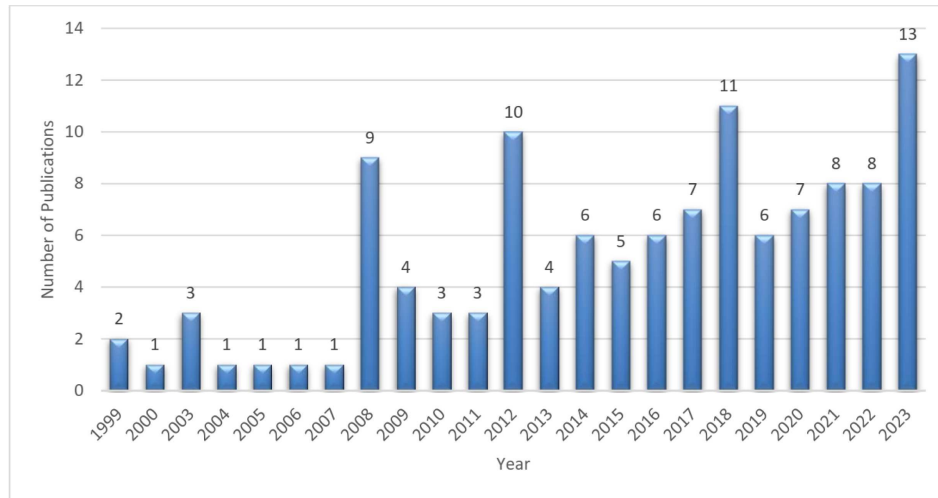


FIGURE 10. Distribution of NWFS articles by year.

TABLE 4. Distribution of articles by journal.

No	Journal	Frequency
1	The International Journal of Advanced Manufacturing Technology	11
2	International Journal of Production Research	9
3	Computers and Operations Research	6
4	Computers and Industrial Engineering	6
5	Applied Soft Computing	5
6	International Journal of Computer Integrated Manufacturing	4
7	Expert Systems with Applications	4
8	IEEE Access	3
9	Omega	3
10	Asia-Pacific Journal of Operational Research	2
11	IEEE Transactions on Industrial Informatics	2
12	Other	65

publications. The analysis also presents clusters of keywords used in the article. The clusters are presented as follows:

- Red Cluster: it includes keywords related to classic NWFS problems. The main keywords such as makespan, batch processing machine, limited machine availability, no-wait constraint, setup time, and total completion time show the focus on process optimization and constraints in NWFS. It indicates that classical NWFS research is still an important and frequently discussed topic in the literature.
- Green Cluster: this cluster categorizes keywords related to flow time minimization in NWFS. Keywords such as heuristic method, m machine NWFS, metaheuristic, non-availability constraint, release date, and flow time highlight the methodologies and approaches used to address this problem. It shows the trend of research focusing on finding efficient solutions for flow time minimization.
- Blue Cluster: in this cluster, keywords such as harmony search, fuzzy due date, greedy algorithm, makespan, multi-objective, and total flowtime focus on multi-objective problems in NWFS. It illustrates the increasing

research interest that tries to solve more than one objective in NWFS, such as balancing efficiency and effectiveness.

- Yellow Cluster: keywords in this cluster, such as batching machine, genetic algorithm, makespan, flexible flow shop, and hybrid flow shop, indicate a focus on hybrid no-wait flow shop problems. It suggests exploring solutions incorporating various aspects of NWFS, integrating traditional and modern approaches.
- Purple Cluster: keywords such as sequence-dependent setup, separable setup times, and server-side constraints indicate a focus on the setup aspect of NWFS. Research in this cluster focuses on optimizing setup times and processes, critical elements in production scheduling.
- Light Blue Cluster: this keyword cluster focuses on energy-efficient, no-wait-flow shops. Keywords such as energy efficient, evolutionary algorithm, and no wait permutation flow shop indicate an increased awareness of the need to make NWFS more environmentally friendly and energy efficient.
- Orange Cluster: it contains keywords such as heuristic, flexible, no-wait flow shop, and no-wait constraint, signifying a flexible approach in NWFS research that combines heuristics and flexibility in scheduling.
- Brown Cluster: focuses on distributed assembly no-wait flow shop and distribution algorithm, indicating exploration of distribution and assembly in the context of NWFS.
- Pink Cluster: includes keywords related to bi-criteria and due date constraint problems, highlighting the complexity and challenges of incorporating multiple factors in the scheduling process.
- Black Cluster: consists of the basic keywords of NWFS research, namely No Wait Flow Shop and makespan, underscoring the core of much research in this area.

Overall, this cluster analysis reveals the diversity of approaches and methods used in NWFS research and the proliferation of new topics that reflect responses to changing technology and industry demands. This evolution signifies the continuous adaptation of the academic community to emerging operational challenges, as well as efforts to integrate considerations such as sustainability and flexibility in the design of production systems.

Overlay Visualization of cluster co-words on NWFS over time is presented in Figure 12. The results show that keywords colored green to yellow, namely “distributed assembly no-wait flow shop scheduling problem (DANWFS)”, “Energy Efficient”, “setup time”, and “no wait flexible flow shop”, are the most dominant and popular topics used in the last five years (2018–2023).

The presence of these keywords in the NWFS research literature marks a shift in research focus towards a more specific and contemporary direction. The keyword “distributed assembly no-wait flow shop scheduling problem” indicates in-depth research into optimization and efficiency in complex production systems, where assembly time and workflow organization are critical. The emphasis on “Energy Efficient” reflects current global research trends focusing on sustainability and energy efficiency, illustrating a response to environmental challenges and the need to reduce carbon footprints in production processes.

Furthermore, the presence of “setup time” as one of the main keywords indicates an increased focus on reducing the time and costs associated with setting up and preparing production processes, which is a key component in improving operational efficiency. “No wait flexible flow shop”, on the other hand, underscores an adaptive approach to production management, where flexibility and the ability to respond to changing demand and market conditions are important.

The results of this study reveal that current NWFS research trends tend towards solving more complex problems relevant to contemporary industrial challenges. The focus on efficiency, sustainability, optimal setup times, and flexibility in workflow management reflects a response to dynamic industry needs and growing awareness of environmental issues. These trends imply that future research will continue to adapt and innovate in the face of increasingly complex operational and environmental challenges.

5. ANALYSIS, TRENDS AND GAPS RESEARCH

This section analyzes published articles, current research trends, and gap identification on NWFS issues. These results are based on articles that researchers have published. The distribution of papers based on the type of NWFS problem is presented in Figure 13. These results show that for most of the NWFS problems,

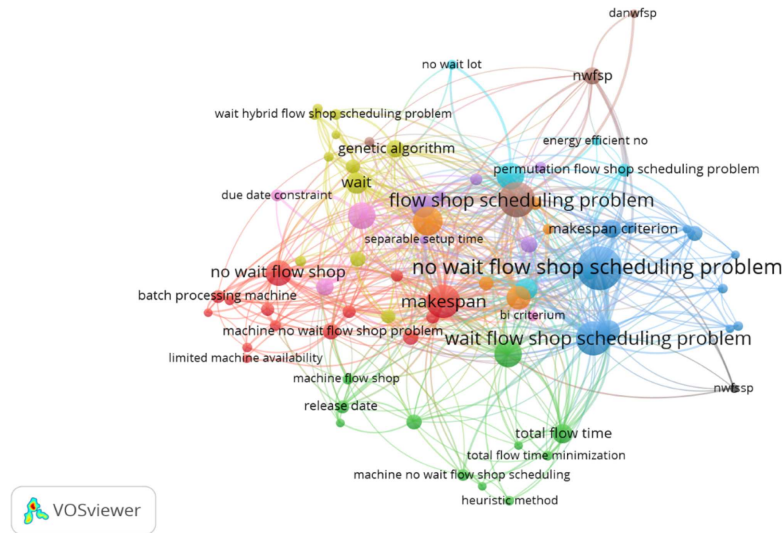


FIGURE 11. Network Visualization of cluster co-words on NWFS.

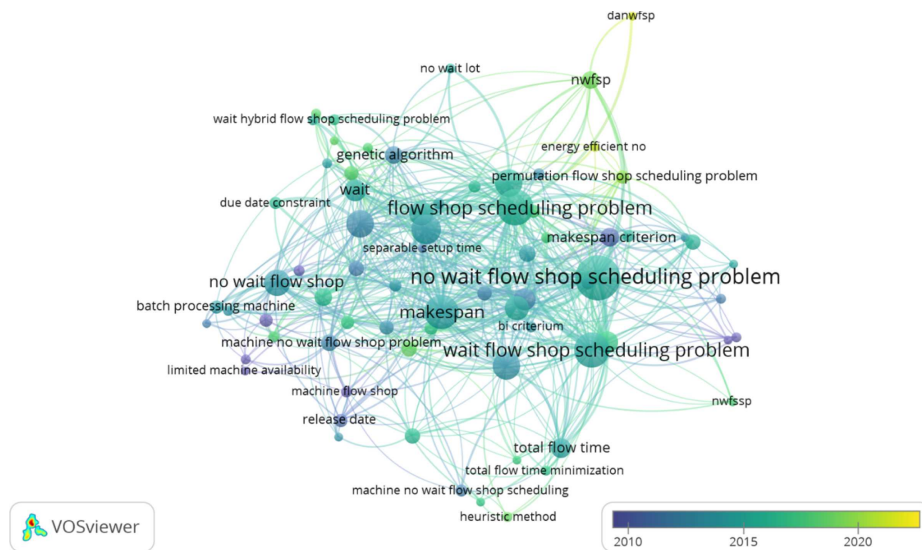


FIGURE 12. Overlay Visualization of cluster co-words on NWFS overtime.

76% (91 articles) address single-objective problems. Meanwhile, only 24% (29 papers) consider solving two or more objective functions simultaneously (multi-objective). In addition, this section also answers the RQ set in Section 2. The description to answer the RQ is presented in the following subsection.

5.1. Analysis variant NWFS

This section describes the analysis of NWFS variants to answer RQ1, namely, what variants dominate NWFS research. The distribution of NWFS problem variants from the published articles is presented in Figure 14. The results show that out of 120 published papers, 97 articles (80.83%) address NWPFS problems. It indicates that

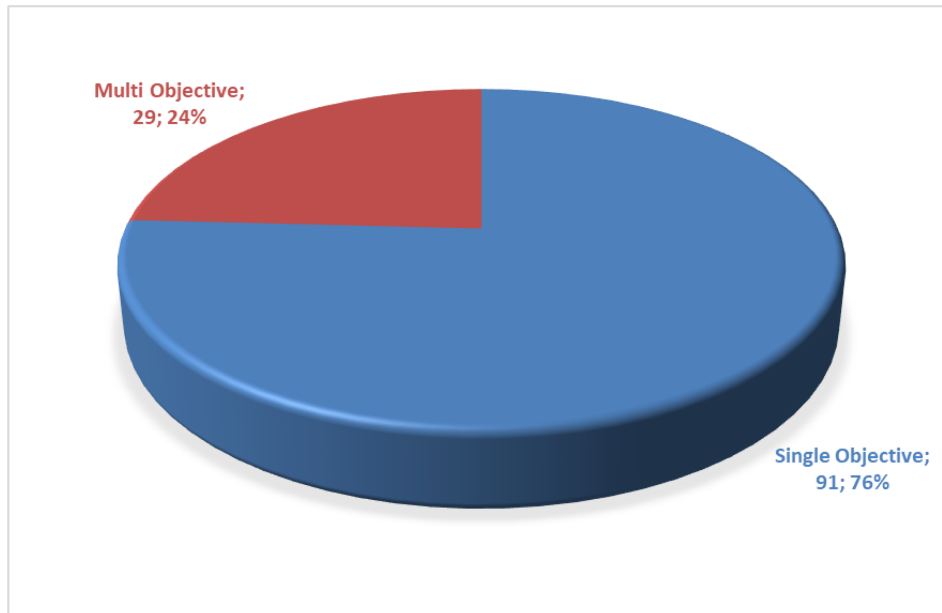


FIGURE 13. Distribution Paper by problem type.

NWPFS is the most frequently used variant in NWFS research. The NWHFS variant came in second place with nine papers. Seven papers were published with the DNWFS variant. The DANWFS variant occupies the fourth position by contributing four papers. Other variants such as MNWFS, DHNWFS, and ANWFS only contributed one paper each.

In the NWPFS problem, the problem for m machines is the most widely investigated problem by researchers [90]. A few NWPFS problems are for solving 2 [43] and three machines [68]. Most of the constraints considered in NWPFS problems are by considering sequence-dependent setup time as in [59, 134, 137, 156]. This constraint represents that the job setup time depends on the job sequence. Other constraints are also proposed, such as involving learning effects [99], Release Time [96, 137], and Uncertain Setup Times [64]. It shows that many researchers offer complex constraints.

5.2. Optimization procedure on NWFS

To answer Research Question 2 (RQ2) procedures that dominate NWFS research, this section outlines the analysis of Optimization procedures on NWFS. Figure 15 presents the Pie chart Distribution of the Optimization procedure on NWFS. The results show that the metaheuristic type of optimization procedure accounts for the largest share of optimization procedures for solving NWFS. Forty-nine articles (41%) proposed metaheuristic procedures with various procedure variants. We note that genetic algorithm procedures are popular algorithms to solve this problem [73, 132, 134]. Genetic algorithm procedures are popular for scheduling cases because they can find near-optimal solutions in complex and often NP-hard problems such as scheduling. Genetic algorithms use the concepts of natural selection, recombination, and mutation, similar to the process of evolution in nature, to generate a population of solutions that continuously evolve from generation to generation. In the context of scheduling, genetic algorithms can generate schedules that satisfy various constraints and preferences, including the assignment of tasks to resources in an efficient manner. The genetic algorithm's adaptability and ability to explore a vast search space makes it a powerful option to tackle complex scheduling challenges. Pseudocode from genetic algorithm can be seen in Figure 16.

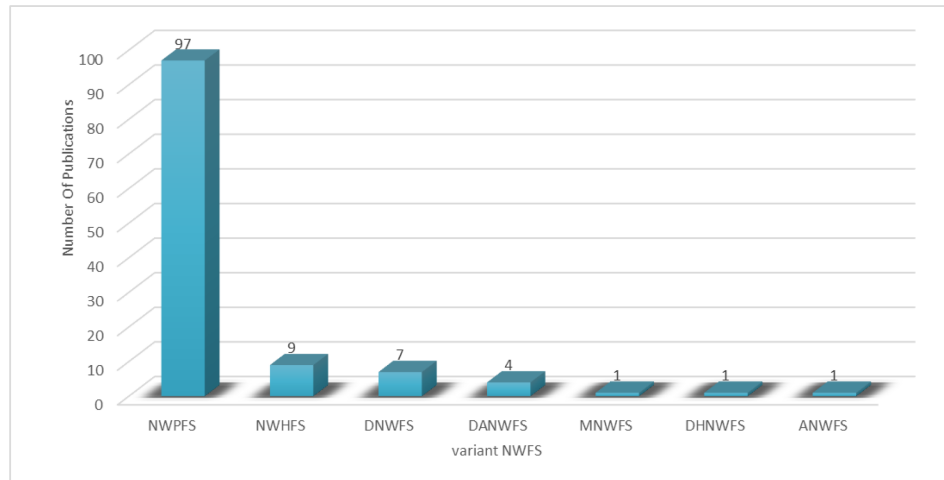


FIGURE 14. Distribution of NWFS problem variants.

Some popular metaheuristic algorithms are particle swarm optimization [76,77,135] and simulated annealing [68,94]. These procedures have the advantage of exploring possible solutions with the computer intelligence embedded in each procedure. We note that this procedure has been popularly used since 2010. Various metaheuristic procedures based on inspiration from nature have been widely applied by researchers recently, such as the Flower pollination algorithm [74], Discrete migrating birds optimization [81], Cuckoo search algorithm [82], and discrete fruit fly optimization algorithm [148].

The hybrid procedure type occupies the second position by contributing 30 papers (25%). A combination of metaheuristic and heuristic procedures dominates most hybrid procedures. Heuristic procedures are mostly used as an initial solution finder in heuristic algorithms. It is used to improve the performance of metaheuristic algorithms. Some of the procedures that have been offered include genetic algorithms and iterated greedy procedures [154], Iterated greedy algorithm and Neighborhood structures Local search [125], and Greedy algorithm, simulated annealing, and variable neighborhood search [116]. Furthermore, hybrid procedures by integrating between metaheuristic procedures are also widely offered, such as particle swarm optimization algorithm and memetic algorithm [115], Variable neighborhood search, Tabu Search, and Simulated Annealing [110], hybrid biogeography-based optimization and variable neighborhood search [107], and Genetic algorithm, artificial bee colony algorithm, migratory bird optimization [147]. These advanced procedures are inseparable from the development of computer technology, which is growing faster.

The heuristic and Exact procedures contributed to 31 and 10 articles, respectively. The heuristic procedure is a popular procedure proposed before 2010. We noted several proposed procedures, such as the algorithm of the Greedy algorithm [36], Gilmore and Gomory [43], and the Heuristic algorithm [47]. These procedures are proposed to find a faster solution even though the quality of the solution is poor. The drawback of heuristic procedures is that they are particularly applicable to specific cases and cannot handle general NWFS cases. Furthermore, Greedy algorithm procedures are popular in scheduling heuristics due to their simple nature but often provide reasonably good solutions relatively quickly. Greedy algorithms in the scheduling context choose the seemingly most favorable action at each step without considering the long-term implications in detail. Although it does not always produce an optimal solution, it is efficient in tackling complex scheduling problems with limited computing time, so it is often used as an initial approach to designing schedules. The pseudocode form of the genetic algorithm can be seen in Figure 17.

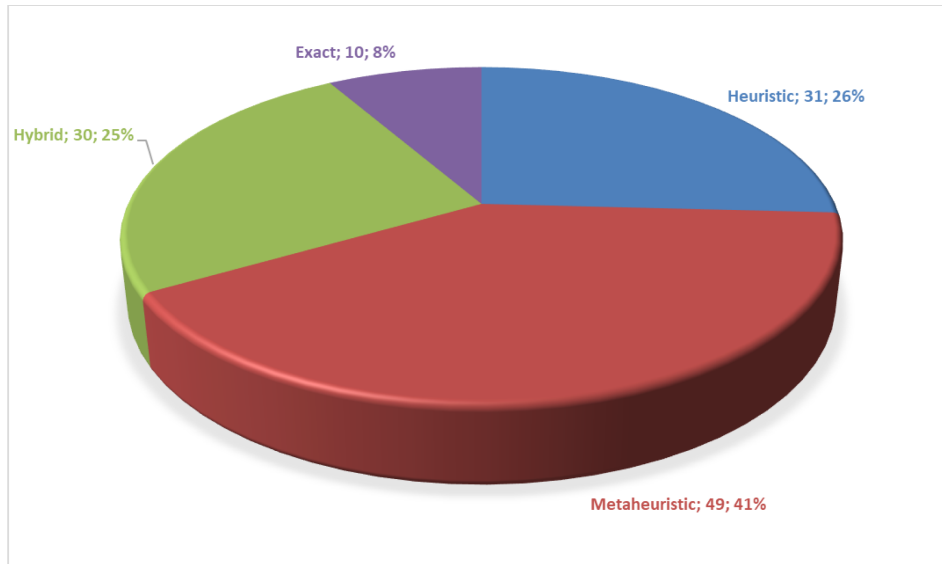


FIGURE 15. Distribution of optimization procedure on NWFS.

Furthermore, regarding the exact procedure, we note that only a few studies have addressed this procedure. Exact methods are inadequate for solving NWFS for medium and large job sizes. The most frequently offered exact procedure is branch and bound [61, 62], which focuses on solving for two machines.

Of the various optimization procedures proposed to solve NWFS problems, algorithm evaluation is a crucial component to determine the effectiveness of various approaches in solving the problem. Algorithm performance is measured based on various aspects, ranging from execution time solution quality to efficiency on a larger scale. The assessment of execution time is critical in applications that require fast response, where efficient algorithms are expected to present solutions in a shorter time. Meanwhile, the solution quality aspect compares the results provided by the algorithm against the optimal or best-known solution, focusing on indicators such as makespan or total solution time. In addition, this research also considers the efficiency of the algorithm in handling larger-scale problems. An ideal algorithm should be able to manage an increase in problem size without experiencing a significant decrease in execution time or solution quality. The stability and consistency of the algorithm were also assessed to determine its ability to produce high-quality solutions consistently, regardless of input variations. The algorithm's ability to handle exceptional cases in NWFS, such as time or resource constraints, is also an essential aspect of the evaluation.

Another approach in this research involves directly comparing the algorithm under study with other existing algorithms to assess differences in performance and efficiency. In addition, a computational cost analysis that considers the computational resources used by the algorithm, such as CPU time and memory, also provides an additional perspective in assessing the algorithm's effectiveness. By measuring the performance improvement of algorithms based on these criteria, researchers can demonstrate the effectiveness of various algorithmic strategies in addressing NWFS challenges, providing a broader view of which approaches are superior in a given context. It helps understand the efficiency of existing algorithms and inspires the development of innovative solutions to future optimization challenges.

5.3. Objective functions of NWFS

This section presents an analysis of the objective functions of the NIFS problem to answer research question 3 (RQ3). The distribution of objective functions on NWFS is shown in Figure 18. The analysis shows that the

Pseudocode form genetic algorithm
Initialize population with random schedules Calculate fitness for each schedule in the population Repeat until convergence or a stopping criterion is met: Select parents for the next generation Create offspring using crossover and mutation Calculate fitness for each offspring Replace the old population with the new population if termination condition met: break Select the best schedule from the final population Return the best schedule as the solution

FIGURE 16. Pseudocode form genetic algorithm.

Pseudocode form Greedy algorithm
Initialize an empty schedule Sort the tasks by a criterion (e.g., earliest deadline first, shortest processing time, etc.) While there are tasks remaining: Select the next task from the sorted list If adding the task to the schedule does not violate any constraints: Add the task to the schedule else: Move to the next task Return the final schedule

FIGURE 17. Pseudocode form Greedy algorithm.

main focus in NWFS research is on makespan optimization (Cmax minimization), with 87 articles (accounting for 58%) concentrating on this aspect. It shows that most research focuses on the classical objective function of minimizing the completion time. In addition, studies lead to other objective functions, such as tardiness minimization and total flow time, represented by 21 and 20 articles, respectively. In conclusion, although there are variations in the objective functions studied, the dominance of research in makespan minimization indicates a strong tendency to prioritize time efficiency in NWFS.

Some studies also tried to solve multi-objective problems with multiple objective functions, such as minimizing makespan and total flow time [127], minimizing makespan and tardiness [133], and minimizing makespan and lateness [154]. Researchers also tried to measure environmental performance with the objective function of energy minimization as in [149, 150] with the objective function of minimizing makespan and total energy consumption. We note that most multi-objective research addresses bi-objective function optimization problems.

5.4. Trend and gaps research

Based on the analysis presented in the previous section, this section presents the Trends and Gaps in Research on NWFS problems. The distribution of papers based on single-objective and multi-objective classification of NWFS for each year is presented in Figure 19. Several significant findings can be discussed based on the bar graph analysis displaying the research trends related to single-objective and multi-objective NWFS issues. There is an overall increasing trend in the number of publications related to NWFS, which indicates the importance of this topic in the field of industry and operations. In the early phase (1999–2007), the dominance of single-

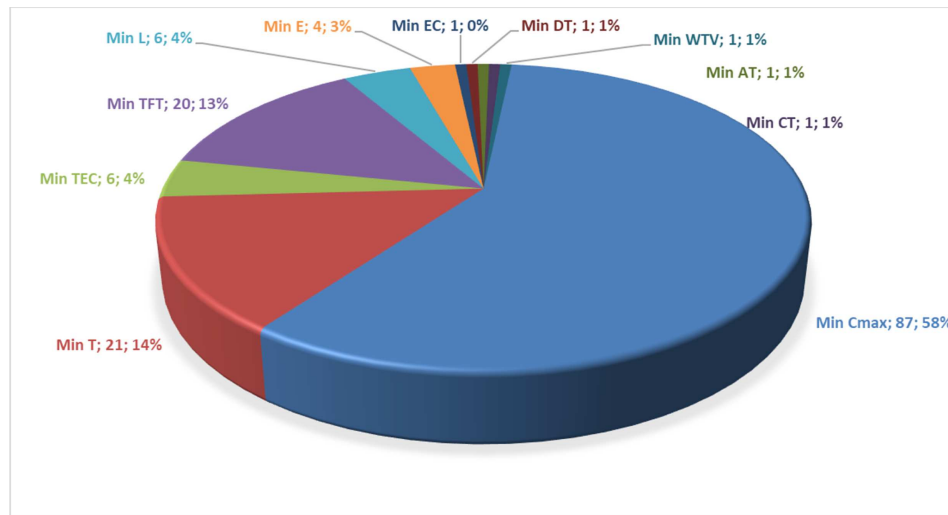


FIGURE 18. Distribution of objective functions on NWFS.

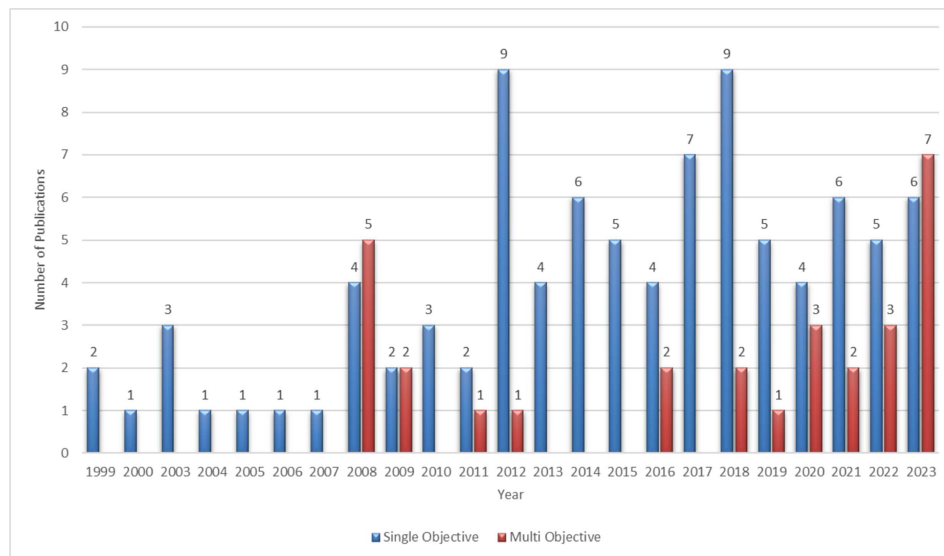


FIGURE 19. Distribution of papers based on single-objective and multi-objective classification of NWFS by year.

objective research reflected the initial focus on solving NWFS problems more simply and directly. However, since 2008, there has been a significant change with the emergence and increase of publications adopting a multi-objective approach.

There has been a spike in the number of multi-objective publications in specific years, particularly in 2014 and 2017, where the number was on par with single-objective publications. It indicates a turning point in NWFS research, where the complexity and variety of objectives in scheduling are becoming increasingly recognized and studied. The years with the highest number of publications, namely 2014, 2017, and 2020 for single objectives and 2014, 2017, and 2023 for multi-objectives, mark essential periods in the evolution of NWFS research. In

Classification	Single-Objective				Multi-Objective			
	Heuristic	Metaheuristic	Hybrid	Exact	Heuristic	Metaheuristic	Hybrid	Exact
NWPFS	[26] [19] [40]	[50] [80] [34]	[79]	[44]	[111]	[109] [110]	[120]	[107]
	[29] [32] [30]	[76] [71] [55]	[81]	[45]	[106]	[113] [117]	[116]	[108]
	[53] [33] [35]	[51] [52] [66]	[90]	[48]	[131]	[114] [122]	[121]	
	[31] [36] [43]	[67] [56] [57]	[85]	[46]	[112]	[115] [138]	[118]	
	[41] [37] [27]	[68] [65] [72]	[87]	[47]	[130]		[119]	
	[42] [70] [86]	[73] [60] [58]	[91]	[66]			[123]	
	[54] [28] [38]	[61] [83] [74]	[96]					
	[39]	[84] [59] [62]	[92]					
		[78] [75] [63]	[93]					
	(22 papers)	[64] [69] [77]	[88]			(7 papers)		(2 papers)
		[49] [84] [85]	[94]		(5 papers)			
		[86]	[89]					
		(34 papers)	[82]	(6 papers)			(6 papers)	
			[95]					
			(15 papers)					
NWHFS	[20] [99]	[102]	[103]	[97]	-	[129]	-	-
	[100]		[101]	[98]		(1 paper)		
	(3 papers)	(1 paper)	(2 paper)	(2 paper)				
DNWFS	[21]	-	[104]	-	[124]	[125] [126]	[127] [128]	-
	(1 paper)		(1 paper)		(1 paper)	(2 paper)	(2 paper)	
MNWFS	[22]	-	-	-	-	-	-	-
	(1 paper)							
DANWFS	[105] [24]	-	-	-	[153]	[130]	-	-
	(2 paper)				(1 paper)	(1 paper)		
DHNWFS	-	[25]	-	-	-	-	-	-
		(1 paper)						
ANWFS			[23]					
			(1 Papers)					

shows the number of researchers

shows limited studies

shows a lack of research

FIGURE 20. NWFS problem gaps map based on the variant, problem type, and optimization procedure.

particular, in 2023, the number of publications with multi-objectives peaked at 7, indicating this approach's importance in current research.

The results show that, in recent years, the trend has been that the number of multi-objective publications has often paralleled or approached the number of single-objective publications. It indicates a shift in research interest, where more complex and holistic approaches to addressing NWFS issues are becoming more desirable. Overall, the graph illustrates the consistent growth in NWFS research and highlights the shift in interest towards more complex and comprehensive multi-objective approaches in recent years.

Furthermore, the research trend involving multiple constraints is also seen in NWFS research. By utilizing more complex constraints, several constraints in the NWFS environment are offered to NWFS research, such as transportation times, learning effects, sequence-dependent setup times, and unequal release dates. Therefore, NWFS problems need to consider multiple constraints at once in investigating NWFS problems.

Metaheuristic procedures based on optimization procedures to solve the NWFS problem have been widely proposed. With the development of advanced computer technology, the research trend that procedures are increasingly significant has been offered recently. Therefore, advanced procedures must be considered to solve the increasingly complex NWFS problems. In addition, based on the objective function, the current study trend encourages researchers to care about the environment, such as energy and emission minimization, so there is insufficient research on this issue.

The NWFS problem gaps map based on the variant, problem type, and optimization procedure is presented in Figure 20. The results show that the variants NWHFS, DNWFS, MNWFS, DANWFS, DHNWFS, and ANWFS need attention in future research. Most papers only solve single-objective problems, and very few studies consider multi-objective simultaneously. It shows that multi-objective research opportunities are up-and-coming in NWFS problems.

Some limitations of previous research are described as follows:

- (1) Research in this area tends to be divided between single-objective and multi-objective optimization approaches. Despite significant progress, there is still room for developing more efficient methods, especially for handling complex multi-objective optimization cases. It suggests that new approaches are needed to balance various objectives, such as time efficiency and energy consumption.
- (2) Various techniques such as exact, heuristic, metaheuristic, and hybrid have been used for optimization. However, challenges arise when facing large solution spaces and high-complexity cases, where achieving an optimal solution becomes more difficult. It emphasizes the need for innovation in optimization techniques to overcome this complexity.
- (3) Moreover, although various NWFS variants have been classified, the practical application of these variants in different industrial scenarios still needs to be further explored. Each variant has assumptions and limitations that must be considered, especially in tailoring solutions to the specific needs of an industry.
- (4) Also, research has generally focused on using heuristics, metaheuristics, and hybrid methods, while exploring exact methods that may provide more optimal solutions is lacking. It shows the importance of diversification in research methodologies to achieve more comprehensive results.
- (5) Regarding the variety of NWFS cases, the literature review showed limitations in exploring specific cases or unique industry conditions. It means there is a need for more in-depth research on the practical application of different NWFS variants, considering specific industry conditions and challenges.
- (6) Another limitation is the influence of metaheuristic algorithm parameters on solution quality. The lack of research suggests the need for experimental design to determine optimal parameter values.
- (7) The main focus has been on classical objective function optimization in the context of objective functions. However, environmental aspects such as minimization of energy consumption and emissions have not received much attention, which suggests room for more sustainability-focused research.
- (8) Another limitation is in addressing the more complex limitations and uncertainties of NWFS. Most studies use classical limitations, such as deterministic processing times. In contrast, more complex limitations and uncertainties have not been studied much.
- (9) Finally, variants such as NWHFS, DNWFS, MNWFS, DANWFS, DHNWFS, and ANWFS show the need for more attention in future research. These limitations open up opportunities for researchers to develop new approaches that are more innovative and effective in solving the NWFS problem.

In the research on NWFS scheduling, several vital areas require further research to overcome the existing challenges and limitations. Some open questions for future work are as follows:

- (1) How can we develop more efficient exact methods for solving NWFS problems that can provide optimal solutions within realistic computational time limits?

- (2) In which industries can variants of NWFS be effectively applied, and how can these models be customized to meet the specific needs of these industries?
- (3) How can more robust multi-objective optimization models be created to address the complex challenges in NWFS, including integrating objectives such as time minimization and energy consumption reduction?
- (4) How can new technologies such as machine learning and artificial intelligence be integrated into NWFS to improve the efficiency and effectiveness of solutions?

6. CONCLUSION AND FUTURE RESEARCH

In this paper, a review of 120 articles related to the NWFS problem has been conducted, focusing on various NWFS variants, optimization procedures, and objective functions. This review is expected to help identify existing research gaps and be a starting point for new research endeavors on the NWFS problem. The findings show that NWFS problems dominate the research in the field of NWFS. In addition, metaheuristic procedures prove to be popular in NWFS research. The most researched and dominating objective function in NWFS research is makespan minimization. It indicates that the main focus of research in this area is on improving the efficiency of the completion time of work schedules in networks.

We also outline future research directions that are important for further exploration in the NWFS context. First, future research should focus on developing methods to solve multi-function objectives in NWFS. It will broaden the research scope and incorporate essential aspects such as efficiency, sustainability, and reliability. Secondly, it is essential to integrate more complex and unique constraints in the NWFS model, including specific operational conditions, resource constraints, and quality requirements. Furthermore, future research directions should involve a deeper exploration of NWFS variants such as NWHFS, DNWFS, MNWFS, DANWFS, DHN-WFS, and ANWFS. It will enable a better understanding of how the various variants can be applied under different conditions and for different purposes. Also, it is essential to research approaches that handle three or more objective functions simultaneously, which can improve the effectiveness and efficiency of the model.

Future research should also focus on NWFS scheduling with environmental considerations, such as energy use, emissions, and waste management, encouraging the development of more sustainable and environmentally friendly models. In addition, research should consider NWFS scheduling under uncertain conditions, an essential reality in real-world applications. We also recommend developing more advanced heuristic and meta-heuristic methods, which can provide optimal solutions to NWFS problems, as well as more in-depth research on the influence of algorithm parameters on solution quality. Developing new methodologies that combine exact and heuristic methods can help achieve optimal solutions more efficiently. Industry-specific case studies can provide insights into how NWFS models can be customized to address specific challenges in a particular industry.

Finally, we encourage research integrating multiple objectives in one optimization model, such as combining production efficiency with environmental sustainability, to create more holistic and sustainable solutions. It will help adapt the NWFS model to the needs of an evolving world that emphasizes sustainability.

Conflict of Interest

All authors have agreed to authorship, read, and approve the manuscript and have given consent for submission and subsequent publication. The authors guarantee that the contribution to the work has not been previously published elsewhere.

Author contribution statement

All authors contributed to the paper's research, writing, and reviewing. Declaration of generative AI and AI-assisted technologies in the writing process. While preparing this work, the author(s) used Grammarly to improve the English language and readability. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

REFERENCES

- [1] A.N.H. Zaied, M.M. Ismail and S.S. Mohamed, Permutation flow shop scheduling problem with makespan criterion: literature review. *J. Theor. Appl. Inf. Technol.* **99** (2021) 830–848.
- [2] H. Öztöp, M.F. Tasgetiren, D.T. Eliyi, Q.-K. Pan and L. Kandiller, An energy-efficient permutation flowshop scheduling problem. *Expert Syst. App.* **150** (2020) 113279.

- [3] Y. Li, X. Li, L. Gao, L. Fu and C. Wang, An efficient critical path based method for permutation flow shop scheduling problem. *J. Manuf. Syst.* **63** (2022) 344–353.
- [4] W. Liao and Y. Fu, Min-max regret criterion-based robust model for the permutation flow-shop scheduling problem. *Eng. Optim.* **52** (2020) 687–700.
- [5] W. Shao, D. Pi and Z. Shao, Local search methods for a distributed assembly no-idle flow shop scheduling problem. *IEEE Syst. J.* **13** (2018) 1945–1956.
- [6] A. Goli and T. Keshavarz, Just-in-time scheduling in identical parallel machine sequence-dependent group scheduling problem. *J. Ind. Manage. Optim.* **18** (2022) 3807.
- [7] D. Yüksel, M.F. Taggetiren, L. Kandiller and L. Gao, An energy-efficient bi-objective no-wait permutation flowshop scheduling problem to minimize total tardiness and total energy consumption. *Comput. Ind. Eng.* **145** (2020) 106431.
- [8] M.A. Baihaqi and D.M. Utama, No-wait flowshop permutation scheduling problem: fire hawk optimizer vs beluga whale optimization algorithm. *Jurnal Ilmiah Teknik Industri* **22** (2023) 124–136.
- [9] T. Aldowaisan and A. Allahverdi, New heuristics for m -machine no-wait flowshop to minimize total completion time. *Omega* **32** (2004) 345–352.
- [10] H. Röck, The three-machine no-wait flow shop is NP-complete. *J. ACM* **31** (1984) 336–345.
- [11] D.A. Rossit, F. Tohmé and M. Frutos, The non-permutation flow-shop scheduling problem: a literature review. *Omega* **77** (2018) 143–153.
- [12] M.M. Yenisey and B. Yagmahan, Multi-objective permutation flow shop scheduling problem: literature review, classification and current trends. *Omega* **45** (2014) 119–135.
- [13] G.M. Komaki, S. Sheikh and B. Malakooti, Flow shop scheduling problems with assembly operations: a review and new trends. *Int. J. Prod. Res.* **57** (2019) 2926–2955.
- [14] E. González-Neira, J. Montoya-Torres and D. Barrera, Flow-shop scheduling problem under uncertainties: review and trends. *Int. J. Ind. Eng. Comput.* **8** (2017) 399–426.
- [15] H.H. Miyata and M.S. Nagano, The blocking flow shop scheduling problem: a comprehensive and conceptual review. *Expert Syst. App.* **137** (2019) 130–156.
- [16] Ö. Tosun, M.K. Marichelvam and N. Tosun, A literature review on hybrid flow shop scheduling. *Int. J. Adv. Oper. Manage.* (2020) 156–194.
- [17] J.S. Neufeld, S. Schulz and U. Buscher, A systematic review of multi-objective hybrid flow shop scheduling. *Eur. J. Oper. Res.* **309** (2023) 1–23.
- [18] D.M. Utama, M.D. Primayesti, S.Z. Umamy, B.M.N. Kholifa and A.D. Yasa, A systematic literature review on energy-efficient hybrid flow shop scheduling. *Cogent Eng.* **10** (2023) 2206074.
- [19] B. Kitchenham and P. Brereton, A systematic review of systematic review process research in software engineering. *Inf. Softw. Technol.* **55** (2013) 2049–2075.
- [20] R. Ojstersek, M. Brezocnik and B. Buchmeister, Multi-objective optimization of production scheduling with evolutionary computation: a review. *Int. J. Ind. Eng. Comput.* **11** (2020) 359–376.
- [21] D.S. Widodo, Improve algoritma hodgson untuk meminimasi jumlah job terlambat pada penjadwalan flow shop. *Jurnal Teknik Teknik Industri* **19** (2018) 73–81.
- [22] A. Goli, A.-M. Golmohammadi and J.-L. Verdegay, Two-echelon electric vehicle routing problem with a developed moth-flame meta-heuristic algorithm. *Oper. Manage. Res.* **15** (2022) 891–912.
- [23] A. Goli, A. Ala and M. Hajiaghahi-Keshteli, Efficient multi-objective meta-heuristic algorithms for energy-aware non-permutation flow-shop scheduling problem. *Expert Syst. App.* **213** (2023) 119077.
- [24] D.M. Utama, D.S. Widodo, M.F. Ibrahim, K. Hidayat, T. Baroto and A. Yurifah, The hybrid whale optimization algorithm: a new metaheuristic algorithm for energy-efficient on flow shop with dependent sequence setup, in International Conference on Science and Technology 2019 Surabaya, Indonesia. IOP Publishing. Vol. 1569 (2020) 022094.
- [25] D.M. Utama, D.S. Widodo, M.F. Ibrahim and S.K. Dewi, An effective hybrid ant lion algorithm to minimize mean tardiness on permutation flow shop scheduling problem. *Int. J. Adv. Intell. Inf.* **6** (2020) 23–35.
- [26] D.S. Widodo and D.M. Utama, The hybrid ant lion optimization flow shop scheduling problem for minimizing completion time, in International Conference on Science and Technology 2019 Surabaya, Indonesia. IOP Publishing. Vol. 1569 (2020) 022097.
- [27] R.L. Graham, E.L. Lawler, J.K. Lenstra and A.H.G.R. Kan, Optimization and approximation in deterministic sequencing and scheduling: a survey, in *Annals of Discrete Mathematics*, edited by P.L. Hammer, E.L. Johnson and B.H. Korte. Vol. 5. Elsevier (1979) 287–326.
- [28] D.M. Utama, A.A.P. Salima and D.S. Widodo, A novel hybrid archimedes optimization algorithm for energy-efficient hybrid flow shop scheduling. *Int. J. Adv. Intell. Inf.* **8** (2022) 237–250.
- [29] D.M. Utama, M.F. Ibrahim, D.S. Wijaya, D.S. Widodo and M.D. Primayesti, A novel hybrid multi-verse optimizer algorithm for energy-efficient permutation flow shop scheduling problem. *J. Phys. Conf. Ser.* **2394** (2022) 012006.
- [30] D.M. Utama, A modified beluga whale optimization for optimizing energy-efficient no-idle permutation flow shop scheduling problem. *Proc. Comput. Sci.* **227** 2023 55–63.
- [31] D.M. Utama, Minimizing number of tardy jobs in flow shop scheduling using a hybrid whale optimization algorithm, in The International Conference on Industrial Automation, Smart Grid and its Application (ICIASGA) 2020, Madiun, Jawa Timur, Indonesia. Vol. 1845. IOP Publishing (2021) 012017.

- [32] I. Amallynda, The discrete particle swarm optimization algorithms for permutation flowshop scheduling problem. *Jurnal Teknik Industri* **20** (2019) 105–116.
- [33] D.R. Wati and I. Amallynda, A no-idle flow shop scheduling using fire hawk optimizer to minimize energy consumption. *Jurnal Teknik Industri* **24** (2023) 65–80.
- [34] Y.M. Risma and D.M. Utama, AVOA and ALO Algorithm for energy-efficient no-idle permutation flow shop scheduling problem: a comparison study. *Jurnal Optimasi Sistem Industri* **22** (2023) 126–141.
- [35] A.N.A.A.K. Jabari and A. Hasan, Energy-aware scheduling in hybrid flow shop using firefly algorithm. *Jurnal Teknik Industri* **22** (2021) 18–30.
- [36] Z. Liu, J. Xie, J. Li and J. Dong, A heuristic for two-stage no-wait hybrid flowshop scheduling with a single machine in either stage. *Tsinghua Sci. Technol.* **8** (2003) 43–48.
- [37] Y. Wang, X. Li, R. Ruiz and S. Sui, An iterated greedy heuristic for mixed no-wait flowshop problems. *IEEE Trans. Cybern.* **48** (2017) 1553–1566.
- [38] A. Mozdgir, S. Fatemi Ghomi, F. Jolai and J. Navaei, Three meta-heuristics to solve the no-wait two-stage assembly flow shop scheduling problem. *Sci. Iran.* **20** (2013) 2275–2283.
- [39] M. Komaki and B. Malakooti, General variable neighborhood search algorithm to minimize makespan of the distributed no-wait flow shop scheduling problem. *Prod. Eng.* **11** (2017) 315–329.
- [40] H. Li, X. Li and L. Gao, A discrete artificial bee colony algorithm for the distributed heterogeneous no-wait flowshop scheduling problem. *Appl. Soft Comput.* **100** (2021) 106946.
- [41] F. Zhao, Z. Xu, L. Wang, N. Zhu, T. Xu and J. Jonrinaldi, A population-based iterated greedy algorithm for distributed assembly no-wait flow-shop scheduling problem. *IEEE Trans. Ind. Inf.* **19** (2023) 6692–6705.
- [42] C.A. Glass, J.N. Gupta and C.N. Potts, Two-machine no-wait flow shop scheduling with missing operations. *Math. Oper. Res.* **24** (1999) 911–924.
- [43] M.-L. Espinouse, P. Formanowicz and B. Penz, Minimizing the makespan in the two-machine no-wait flow-shop with limited machine availability. *Comput. Ind. Eng.* **37** (1999) 497–500.
- [44] D. Li, K. Chen and X. Wang, Properties of two-machine no-wait flow-shop scheduling with a non-resumable unavailable interval. *J. Ind. Prod. Eng.* **34** (2017) 232–238.
- [45] K. Chen, D. Li and X. Wang, Makespan minimization in two-machine flow-shop scheduling under no-wait and deterministic unavailable interval constraints. *J. Syst. Sci. Syst. Eng.* **29** (2020) 400–411.
- [46] M.-L. Espinouse, P. Formanowicz and B. Penz, Complexity results and approximation algorithms for the two machine no-wait flow-shop with limited machine availability. *J. Oper. Res. Soc.* **52** (2001) 116–121.
- [47] M.A. Kubzin and V.A. Strusevich, Two-machine flow shop no-wait scheduling with a nonavailability interval. *Nav. Res. Logistics* **51** (2004) 613–631.
- [48] R. Shafaei, M. Rabiee and M. Mirzaeyan, An adaptive neuro fuzzy inference system for makespan estimation in multiprocessor no-wait two stage flow shop. *Int. J. Comput. Integr. Manuf.* **24** (2011) 888–899.
- [49] M. Sviridenko, Makespan minimization in no-wait flow shops: a polynomial time approximation scheme. *SIAM J. Discrete Math.* **16** (2003) 313–322.
- [50] X. Li and C. Wu, Heuristic for no-wait flow shops with makespan minimization based on total idle-time increments. *Sci. Chin. Ser. F: Inf. Sci.* **51** (2008) 896.
- [51] Q.-K. Pan, L. Wang and B.-H. Zhao, An improved iterated greedy algorithm for the no-wait flow shop scheduling problem with makespan criterion. *Int. J. Adv. Manuf. Technol.* **38** (2008) 778–786.
- [52] D. Laha and U.K. Chakraborty, A constructive heuristic for minimizing makespan in no-wait flow shop scheduling. *Int. J. Adv. Manuf. Technol.* **41** (2009) 97–109.
- [53] H. Yuan, Y. Jing, J. Huang and T. Ren, Optimal research and numerical simulation for scheduling no-wait flow shop in steel production. *J. Appl. Math.* **2013** (2013). DOI: [10.1155/2013/498282](https://doi.org/10.1155/2013/498282).
- [54] K. Nailwal, D. Gupta and K. Jeet, Heuristics for no-wait flow shop scheduling problem. *Int. J. Ind. Eng. Comput.* **7** (2016) 671–680.
- [55] F. Zhao, X. He, Y. Zhang, W. Lei, W. Ma, C. Zhang and H. Song, A jigsaw puzzle inspired algorithm for solving large-scale no-wait flow shop scheduling problems. *Appl. Intell.* **50** (2020) 87–100.
- [56] F.S.D. Almeida and M.S. Nagano, Heuristics to optimize total completion time subject to makespan in no-wait flow shops with sequence-dependent setup times. *J. Oper. Res. Soc.* **74** (2023) 362–373.
- [57] E. Bertolissi, Heuristic algorithm for scheduling in the no-wait flow-shop. *J. Mater. Process. Technol.* **107** (2000) 459–465.
- [58] D. Laha, J.N. Gupta and S.U. Sapkal, A penalty-shift-insertion-based algorithm to minimize total flow time in no-wait flow shops. *J. Oper. Res. Soc.* **65** (2014) 1611–1624.
- [59] X. Li, Z. Yang, R. Ruiz, T. Chen and S. Sui, An iterated greedy heuristic for no-wait flow shops with sequence dependent setup times, learning and forgetting effects. *Inf. Sci.* **453** (2018) 408–425.
- [60] G. Liu, S. Song and C. Wu, Some heuristics for no-wait flowshops with total tardiness criterion. *Comput. Oper. Res.* **40** (2013) 521–525.
- [61] L.-H. Su and Y.-Y. Lee, The two-machine flowshop no-wait scheduling problem with a single server to minimize the total completion time. *Comput. Oper. Res.* **35** (2008) 2952–2963.
- [62] F. Ben Chihaoui, I. Kacem, A.B. Hadj-Alouane, N. Dridi and N. Rezg, No-wait scheduling of a two-machine flow-shop to minimise the makespan under non-availability constraints and different release dates. *Int. J. Prod. Res.* **49** (2011) 6273–6286.

- [63] M. Labidi, A. Kooli, T. Ladhari, A. Gharbi and U.S. Suryahatmaja, A computational study of the two-machine no-wait flow shop scheduling problem subject to unequal release dates and non-availability constraints. *IEEE Access* **6** (2018) 16294–16304.
- [64] M. Allahverdi, An improved algorithm to minimize the total completion time in a two-machine no-wait flow-shop with uncertain setup times. *J. Project Manage.* **7** (2022) 1–12.
- [65] H. Samarghandi and M. Behroozi, On the exact solution of the no-wait flow shop problem with due date constraints. *Comput. Oper. Res.* **81** (2017) 141–159.
- [66] C. Koulamas and G.J. Kyparisis, Two-stage no-wait proportionate flow shop scheduling with minimal service time variation and optional job rejection. *Eur. J. Oper. Res.* **305** (2023) 608–616.
- [67] A. Azerine, M. Boudhar and D. Rebaine, A two-machine no-wait flow shop problem with two competing agents. *J. Comb. Optim.* **43** (2022) 168–199.
- [68] T.-Y. Wang, Y.-H. Yang and H.-J. Lin, Comparison of scheduling efficiency in two/three-machine no-wait flow shop problem using simulated annealing and genetic algorithm. *Asia-Pac. J. Oper. Res.* **23** (2006) 41–59.
- [69] S. Muthuswamy, M.C. Vélez-Gallego, J. Maya and M. Rojas-Santiago, Minimizing makespan in a two-machine no-wait flow shop with batch processing machines. *Int. J. Adv. Manuf. Technol.* **63** (2012) 281–290.
- [70] K.-W. Pang, A genetic algorithm based heuristic for two machine no-wait flowshop scheduling problems with class setup times that minimizes maximum lateness. *Int. J. Prod. Econ.* **141** (2013) 127–136.
- [71] J. Grabowski and J. Pempera, Some local search algorithms for no-wait flow-shop problem with makespan criterion. *Comput. Oper. Res.* **32** (2005) 2197–2212.
- [72] F. Zhao, F. Xue, G. Yang, W. Ma, C. Zhang and H. Song, A fitness landscape analysis for the no-wait flow shop scheduling problem with factorial representation. *IEEE Access* **7** (2019) 21032–21047.
- [73] I.A. Chaudhry and A.M. Khan, Minimizing makespan for a no-wait flowshop using genetic algorithm. *Sadhana* **37** (2012) 695–707.
- [74] G. Deng, M. Wei, Q. Su and M. Zhao, An effective co-evolutionary quantum genetic algorithm for the no-wait flow shop scheduling problem. *Adv. Mech. Eng.* **7** (2015) 1687814015622900.
- [75] H. Samarghandi, A particle swarm optimisation for the no-wait flow shop problem with due date constraints. *Int. J. Prod. Res.* **53** (2015) 2853–2870.
- [76] L.A. Bewoor, V.C. Prakash and S.U. Sapkal, Comparative analysis of metaheuristic approaches for make span minimization for no wait flow shop scheduling problem. *Int. J. Electr. Comput. Eng.* **7** (2017) 417.
- [77] L.A. Bewoor, V. Chandraprakash and S. Sapkal, Evolutionary hybrid particle swarm optimization algorithm to minimize makespan to schedule a flow shop with no wait. *J. Eng. Sci. Technol.* **14** (2019) 609–628.
- [78] W. Shao, D. Pi and Z. Shao, An extended teaching-learning based optimization algorithm for solving no-wait flow shop scheduling problem. *Appl. Soft Comput.* **61** (2017) 193–210.
- [79] F. Zhao, L. Zhang, H. Liu, Y. Zhang, W. Ma, C. Zhang and H. Song, An improved water wave optimization algorithm with the single wave mechanism for the no-wait flow-shop scheduling problem. *Eng. Optim.* **51** (2019) 1727–1742.
- [80] H. Zhu, X. Qi, F. Chen, X. He, L. Chen and Z. Zhang, Quantum-inspired cuckoo co-search algorithm for no-wait flow shop scheduling. *Appl. Intell.* **49** (2019) 791–803.
- [81] S. Zhang, X. Gu and F. Zhou, An improved discrete migrating birds optimization algorithm for the no-wait flow shop scheduling problem. *IEEE Access* **8** (2020) 99380–99392.
- [82] H. Zhu, N. Luo and X. Li, A quantum-inspired cuckoo co-evolutionary algorithm for no-wait flow shop scheduling. *IET Collab. Intell. Manuf.* **3** (2021) 105–118.
- [83] W. Shao, D. Pi and Z. Shao, A hybrid discrete optimization algorithm based on teaching-probabilistic learning mechanism for no-wait flow shop scheduling. *Knowl.-Based Syst.* **107** (2016) 219–234.
- [84] S. Zhang and X. Gu, A discrete whale optimization algorithm for the no-wait flow shop scheduling problem. *Meas. Control* **56** (2023) 1764–1779.
- [85] J. Li, X. Guo and Q. Zhang, Multi-strategy discrete teaching–learning-based optimization algorithm to solve no-wait flow-shop-scheduling problem. *Symmetry* **15** (2023) 1430.
- [86] J. Yu, H. Zhang and Y. Dong, Research on No-wait flow shop scheduling based on discrete state transition algorithm. *J. Syst. Simul.* **35** (2023) 1034–1045.
- [87] M.S. Nagano, A.A. Da Silva and L.A.N. Lorena, An evolutionary clustering search for the no-wait flow shop problem with sequence dependent setup times. *Expert Syst. App.* **41** (2014) 3628–3633.
- [88] H. Samarghandi and T.Y. ElMekkawy, Solving the no-wait flow-shop problem with sequence-dependent set-up times. *Int. J. Comput. Integr. Manuf.* **27** (2014) 213–228.
- [89] H. Samarghandi, Studying the effect of server side-constraints on the makespan of the no-wait flow-shop problem with sequence-dependent set-up times. *Int. J. Prod. Res.* **53** (2015) 2652–2673.
- [90] H. Sun, A. Jiang, D. Ge, X. Zheng and F. Gao, A chance constrained programming approach for no-wait flow shop scheduling problem under the interval-valued fuzzy processing time. *Processes* **9** (2021) 789.
- [91] I.A. Chaudhry, I.A. Elbadawi, M. Usman and M. Tajammal Chughtai, Minimising total flowtime in a no-wait flow shop (NWFS) using genetic algorithms. *Ingeniería Investigación* **38** (2018) 68–79.
- [92] K.-Z. Gao, Q.-K. Pan and J.-Q. Li, Discrete harmony search algorithm for the no-wait flow shop scheduling problem with total flow time criterion. *Int. J. Adv. Manuf. Technol.* **56** (2011) 683–692.
- [93] X. Qi, H. Wang, H. Zhu, J. Zhang, F. Chen and J. Yang, Fast local neighborhood search algorithm for the no-wait flow shop scheduling with total flow time minimization. *Int. J. Prod. Res.* **54** (2016) 4957–4972.

- [94] L.A. Bewoor, V. Chandra Prakash and S.U. Sapkal, Evolutionary hybrid particle swarm optimization algorithm for solving NP-hard no-wait flow shop scheduling problems. *Algorithms* **10** (2017) 121.
- [95] C. Qu, Y. Fu, Z. Yi and J. Tan, Solutions to no-wait flow shop scheduling problem using the flower pollination algorithm based on the hormone modulation mechanism. *Complexity* **2018** (2018). DOI: [10.1155/2018/1973604](https://doi.org/10.1155/2018/1973604).
- [96] R. Hu, X. Wu, B. Qian, J.L. Mao and H.P. Jin, An enhanced differential evolution algorithm with fast evaluating strategies for TWT-NFSP with SSTs and RTs. *Complexity* **2020** (2020) 1–11.
- [97] C. Wang, X. Li and Q. Wang, Tabu search for no-wait flowshop scheduling problem to minimize maximum lateness. *J. Southeast Univ.* **26** (2010) 26–30.
- [98] C. Smutnicki, J. Pempera, G. Bocewicz and Z. Banaszak, Cyclic flow-shop scheduling with no-wait constraints and missing operations. *Eur. J. Oper. Res.* **302** (2022) 39–49.
- [99] F. Nouri, S. Samadzad and J. Nahr, Meta-heuristics algorithm for two-machine no-wait flow-shop scheduling problem with the effects of learning. *Uncertain Supply Chain Manage.* **7** (2019) 599–618.
- [100] B. Liu, L. Wang and Y.-H. Jin, An effective hybrid particle swarm optimization for no-wait flow shop scheduling. *Int. J. Adv. Manuf. Technol.* **31** (2007) 1001–1011.
- [101] Q.-K. Pan, L. Wang, M.F. Tasgetiren and B.-H. Zhao, A hybrid discrete particle swarm optimization algorithm for the no-wait flow shop scheduling problem with makespan criterion. *Int. J. Adv. Manuf. Technol.* **38** (2008) 337–347.
- [102] B. Qian, L. Wang, R. Hu, D. Huang and X. Wang, A DE-based approach to no-wait flow-shop scheduling. *Comput. Ind. Eng.* **57** (2009) 787–805.
- [103] J.-Y. Ding, S. Song, J.N. Gupta, R. Zhang, R. Chiong and C. Wu, An improved iterated greedy algorithm with a Tabu-based reconstruction strategy for the no-wait flowshop scheduling problem. *Appl. Soft Comput.* **30** (2015) 604–613.
- [104] O. Engin and A. Güçlü, A new hybrid ant colony optimization algorithm for solving the no-wait flow shop scheduling problems. *Appl. Soft Comput.* **72** (2018) 166–176.
- [105] F. Zhao, H. Liu, Y. Zhang, W. Ma and C. Zhang, A discrete water wave optimization algorithm for no-wait flow shop scheduling problem. *Expert Syst. App.* **91** (2018) 347–363.
- [106] H. Samarghandi and T.Y. ElMekkawy, A meta-heuristic approach for solving the no-wait flow-shop problem. *Int. J. Prod. Res.* **50** (2012) 7313–7326.
- [107] F. Zhao, S. Qin, Y. Zhang, W. Ma, C. Zhang and H. Song, A hybrid biogeography-based optimization with variable neighborhood search mechanism for no-wait flow shop scheduling problem. *Expert Syst. App.* **126** (2019) 321–339.
- [108] M. Rabiee, M. Zandieh and A. Jafarian, Scheduling of a no-wait two-machine flow shop with sequence-dependent setup times and probable rework using robust meta-heuristics. *Int. J. Prod. Res.* **50** (2012) 7428–7446.
- [109] K.-C. Ying, Z.-J. Lee, C.-C. Lu and S.-W. Lin, Metaheuristics for scheduling a no-wait flowshop manufacturing cell with sequence-dependent family setups. *Int. J. Adv. Manuf. Technol.* **58** (2012) 671–682.
- [110] M. Jabbari and V. Azizi, Lot streaming in no-wait multi product flowshop considering sequence dependent setup times and position based learning factors. *Int. J. Eng.* **28** (2015) 1031–1039.
- [111] C. Wang, X. Li and Q. Wang, Accelerated tabu search for no-wait flowshop scheduling problem with maximum lateness criterion. *Eur. J. Oper. Res.* **206** (2010) 64–72.
- [112] T. Aldowaisan and A. Allahverdi, Minimizing total tardiness in no-wait flowshops. *Found. Comput. Decis. Sci.* **37** (2012) 149.
- [113] T.A. Aldowaisan and A. Allahverdi, No-wait flowshop scheduling problem to minimize the number of tardy jobs. *Int. J. Adv. Manuf. Technol.* **61** (2012) 311–323.
- [114] K.-Z. Gao, Q.-K. Pan, J.-Q. Li, Y.-T. Wang and J. Liang, A hybrid harmony search algorithm for the no-wait flow-shop scheduling problems. *Asia-Pac. J. Oper. Res.* **29** (2012) 1250012.
- [115] M. Akhshabi, R. Tavakkoli-Moghaddam and F. Rahnamay-Roodposhti, A hybrid particle swarm optimization algorithm for a no-wait flow shop scheduling problem with the total flow time. *Int. J. Adv. Manuf. Technol.* **70** (2014) 1181–1188.
- [116] M. Tonizza Pereira and M. Seido Nagano, Hybrid metaheuristics for the integrated and detailed scheduling of production and delivery operations in no-wait flow shop systems. *Comput. Ind. Eng.* **170** (2022) 108255.
- [117] M.S. Nagano, A.A. Da Silva and L.A.N. Lorena, A new evolutionary clustering search for a no-wait flow shop problem with set-up times. *Eng. App. Artif. Intell.* **25** (2012) 1114–1120.
- [118] S. Wang, M. Liu and C. Chu, A branch-and-bound algorithm for two-stage no-wait hybrid flow-shop scheduling. *Int. J. Prod. Res.* **53** (2015) 1143–1167.
- [119] M.-N. Azaiez, A. Gharbi, I. Kacem, Y. Makhoulouf and M. Masmoudi, Two-stage no-wait hybrid flow shop with inter-stage flexibility for operating room scheduling. *Comput. Ind. Eng.* **168** (2022) 108040.
- [120] W. Zhong and Y. Shi, Two-stage no-wait hybrid flowshop scheduling with inter-stage flexibility. *J. Comb. Optim.* **35** (2018) 108–125.
- [121] J. Dong, H. Pan, C. Ye, W. Tong and J. Hu, No-wait two-stage flowshop problem with multi-task flexibility of the first machine. *Inf. Sci.* **544** (2021) 25–38.
- [122] B. Naderi, M. Khalili and A.A. Khamseh, Mathematical models and a hunting search algorithm for the no-wait flowshop scheduling with parallel machines. *Int. J. Prod. Res.* **52** (2014) 2667–2681.
- [123] M. Rabiee, F. Jolai, H. Asefi, P. Fattahi and S. Lim, A biogeography-based optimisation algorithm for a realistic no-wait hybrid flow shop with unrelated parallel machines to minimise mean tardiness. *Int. J. Comput. Integr. Manuf.* **29** (2016) 1007–1024.

- [124] H. Xuan, Q. Zheng, B. Li and X. Wang, A novel genetic simulated annealing algorithm for no-wait hybrid flowshop problem with unrelated parallel machines. *ISIJ Int.* **61** (2021) 258–268.
- [125] W. Shao, D. Pi and Z. Shao, Optimization of makespan for the distributed no-wait flow shop scheduling problem with iterated greedy algorithms. *Knowl.-Based Syst.* **137** (2017) 163–181.
- [126] F. Zhao, J. Zhao, L. Wang and J. Tang, An optimal block knowledge driven backtracking search algorithm for distributed assembly no-wait flow shop scheduling problem. *Appl. Soft Comput.* **112** (2021) 107750.
- [127] D. Laha and J.N. Gupta, A Hungarian penalty-based construction algorithm to minimize makespan and total flow time in no-wait flow shops. *Comput. Ind. Eng.* **98** (2016) 373–383.
- [128] N. Madhushini and C. Rajendran, Branch-and-bound algorithms for scheduling in an m -machine no-wait flowshop. *Sādhanā* **45** (2020) 1–11.
- [129] J. Schaller and J.M. Valente, Minimizing total earliness and tardiness in a nowait flow shop. *Int. J. Prod. Econ.* **224** (2020) 107542.
- [130] F.S. de Almeida and M.S. Nagano, An efficient iterated greedy algorithm for a multi-objective no-wait flow shop problem with sequence dependent setup times. *4OR – Q. J. Oper. Res.* (2023). DOI: [10.1007/s10288-023-00535-7](https://doi.org/10.1007/s10288-023-00535-7).
- [131] R. Tavakkoli-Moghaddam, A. Rahimi-Vahed and A. Mirzaei, Solving a multi-objective no-wait flow shop scheduling problem with an immune algorithm. *Int. J. Adv. Manuf. Technol.* **36** (2008) 969–981.
- [132] A. Rahimi-Vahed, B. Javadi, M. Rabbani and R. Tavakkoli-Moghaddam, A multi-objective scatter search for a bi-criteria no-wait flow shop scheduling problem. *Eng. Optim.* **40** (2008) 331–346.
- [133] M. Khalili, Multi-objective no-wait hybrid flowshop scheduling problem with transportation times. *Int. J. Comput. Sci.* **7** (2012) 147–154.
- [134] A. Guevara-Guevara, V. Gómez-Fuentes, L. Posos-Rodríguez, N. Remolina-Gómez and E. González-Neira, Earliness/tardiness minimization in a no-wait flow shop with sequence-dependent setup times. *J. Project Manage.* **7** (2022) 177–190.
- [135] Q.-K. Pan, M.F. Tasgetiren and Y.-C. Liang, A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem. *Comput. Oper. Res.* **35** (2008) 2807–2839.
- [136] V. Riahi and M. Kazemi, A new hybrid ant colony algorithm for scheduling of no-wait flowshop. *Oper. Res.* **18** (2018) 55–74.
- [137] B. Qian, Z.Q. Zhang, R. Hu, H.P. Jin and J.B. Yang, A matrix-cube-based estimation of distribution algorithm for no-wait flow-shop scheduling with sequence-dependent setup times and release times. *IEEE Trans. Syst. Man Cybern. Syst.* **53** (2023) 1492–1503.
- [138] I. Karacan, O. Senvar and S. Bulkan, A novel parallel simulated annealing methodology to solve the no-wait flow shop scheduling problem with earliness and tardiness objectives. *Processes* **11** (2023) 454.
- [139] B. Qian, L. Wang, D.-X. Huang and X. Wang, Multi-objective no-wait flow-shop scheduling with a memetic algorithm based on differential evolution. *Soft Comput.* **13** (2009) 847–869.
- [140] A. Allahverdi, H. Aydilek and A. Aydilek, No-wait flowshop scheduling problem with two criteria; total tardiness and makespan. *Eur. J. Oper. Res.* **269** (2018) 590–601.
- [141] Q.-K. Pan, L. Wang and B. Qian, A novel differential evolution algorithm for bi-criteria no-wait flow shop scheduling problems. *Comput. Oper. Res.* **36** (2009) 2498–2511.
- [142] B. Jarbouli, M. Eddaly and P. Siarry, A hybrid genetic algorithm for solving no-wait flowshop scheduling problems. *Int. J. Adv. Manuf. Technol.* **54** (2011) 1129–1143.
- [143] X.-P. Liao, Y.-G. Liu and X.-P. Li, Hybrid evolutionary algorithm for no-wait flow shops to minimize makespan and total flowtime. *J. Southeast Univ.* **24** (2008) 450–454.
- [144] K. Keskin and O. Engin, A hybrid genetic local and global search algorithm for solving no-wait flow shop problem with bi criteria. *SN Appl. Sci.* **3** (2021) 1–15.
- [145] A.C.L. Gomes, M.G. Ravetti and E.G. Carrano, Multi-objective matheuristic for minimization of total tardiness and energy costs in a steel industry heat treatment line. *Comput. Ind. Eng.* **151** (2021) 106929.
- [146] W. Shao, D. Pi and Z. Shao, A Pareto-based estimation of distribution algorithm for solving multiobjective distributed no-wait flow-shop scheduling problem with sequence-dependent setup time. *IEEE Trans. Autom. Sci. Eng.* **16** (2019) 1344–1360.
- [147] K. Allali, S. Aqil and J. Belabid, Distributed no-wait flow shop problem with sequence dependent setup time: optimization of makespan and maximum tardiness. *Simul. Modell. Pract. Theory* **116** (2022) 102455.
- [148] N. Zhu, F. Zhao, L. Wang, R. Ding, T. Xu and Jonrinaldi, A discrete learning fruit fly algorithm based on knowledge for the distributed no-wait flow shop scheduling with due windows. *Expert Syst. App.* **198** (2022) 116921.
- [149] F. Zhao, T. Jiang and L. Wang, A reinforcement learning driven cooperative meta-heuristic algorithm for energy-efficient distributed no-wait flow-shop scheduling with sequence-dependent setup time. *IEEE Trans. Ind. Inf.* **19** (2023) 8427–8440.
- [150] Q.-Q. Zeng, J.-Q. Li, R.-H. Li, T.-H. Huang, Y.-Y. Han and H.-Y. Sang, Improved NSGA-II for energy-efficient distributed no-wait flow-shop with sequence-dependent setup time. *Complex Intell. Syst.* **9** (2023) 825–849.
- [151] Y. Qin and H. Zhang, Elite particle swarm optimization algorithm for solving the bi-criteria no-wait flexible flow shop problem. *Int. J. Grid Distrib. Comput.* **9** (2016) 267–232.
- [152] F. Zhao, X. Hu, L. Wang, T. Xu, N. Zhu and Jonrinaldi, A reinforcement learning-driven brain storm optimisation algorithm for multi-objective energy-efficient distributed assembly no-wait flow shop scheduling problem. *Int. J. Prod. Res.* **61** (2023) 2854–2872.
- [153] F. Zhao, Z. Xu, X. Hu, T. Xu, N. Zhu and Jonrinaldi, An improved iterative greedy athm for energy-efficient distributed assembly no-wait flow-shop scheduling problem. *Swarm Evol. Comput.* **81** (2023) 101355.

- [154] R. Ruiz and A. Allahverdi, New heuristics for no-wait flow shops with a linear combination of makespan and maximum lateness. *Int. J. Prod. Res.* **47** (2009) 5717–5738.
- [155] X. Wu and A. Che, Energy-efficient no-wait permutation flow shop scheduling by adaptive multi-objective variable neighborhood search. *Omega* **94** (2020) 102117.
- [156] F.S.D. Almeida and M.S. Nagano, Heuristics to optimize total completion time subject to makespan in no-wait flow shops with sequence-dependent setup times. *J. Oper. Res. Soc.* **74** (2023) 362–373.



Please help to maintain this journal in open access!

This journal is currently published in open access under the Subscribe to Open model (S2O). We are thankful to our subscribers and supporters for making it possible to publish this journal in open access in the current year, free of charge for authors and readers.

Check with your library that it subscribes to the journal, or consider making a personal donation to the S2O programme by contacting subscribers@edpsciences.org.

More information, including a list of supporters and financial transparency reports, is available at <https://edpsciences.org/en/subscribe-to-open-s2o>.