Int. J. Res. Ind. Eng. Vol. 7, No. 3 (2018) 254–306

International Journal of Research in Industrial Engineering

www.riejournal.com

Multi-objective Sequence Dependent Setup Times Hybrid Flowshop Scheduling: A Literature Review

F. Ghassemi Tari*, M. Rezapour Niari

Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran.

A B S T R A C T

Multi-criteria sequence dependent setup times scheduling problems exist almost everywhere in real modern manufacturing world environments. Among them, Sequence Dependent Setup Times-Multi-Objective Hybrid Flowshop Scheduling Problem (SDST-MOHFSP) has been an intensifying attention of researchers and practitioners in the last three decades. In this paper, we briefly summarized and classified the current standing of SDST-MOHFSP. All publications are categorized regarding the solution methods, as well as the structure of the hybrid flowshop which helps researcher and practitioner to use/modify proper solution algorithm for solving their specific problem. Furthermore, based on the review of the existing papers, the need for future research is recognized. Accordingly, by recognizing the research gaps, a large number of recommendations for further study have been proposed.

Keywords: Multi-objective algorithms, Hybrid flowshop scheduling, Sequence dependent setup times, Exact methods, Heuristic and metaheuristic algorithms, Literature review.

Article history: *Received: 21 June 2018* *Revised: 01 October 2018* *Accepted: 30 November 2018*

1. Introduction

Flow pattern of scheduling can be classified based on the required number of operations to process a job, the number of available machines per operation, and number of work centers. Based on machines, the concept of routing can be categorized as single machine, flowshop, jobshop, and open shop scheduling problems. Considering processing stages instead of machines, the concept of routing is introduced as parallel machine shop, and hybrid flowshop scheduling problems. The process industry such as automotive [4], electronics [182], glass [92], and textile [80] industries can be modeled as a hybrid flowshop.

The Hybrid Flowshop Scheduling Problem (HFSP) has received considerable attention in the scheduling literature. Among those, the flowshop scheduling problem with Sequence Dependent Setup Times (SDST) has been one of the most well-known problems [6, 8, 11].

* Corresponding author
E-mail address: ghasemi@sharif.edu
DOI: 10.22105/riej.2018.145136.1057

Despite this fact, most research studies on SDST-HFSP have been focused on single objective problems which are believed to be lacking for the majority of the real-world problems. However, according to our knowledge, few papers have been published dealing with the optimization of hybrid flowshop setup time with more than one objective at the same time.

Minimization of makespan and minimization of flow time are considered as the most popular measures of performance in multi-objective flowshop scheduling problems [134]. While these objectives assess system utilization, the due date related objectives measure the realization of the customer satisfaction [122]. To have a more fruitful and a more effective production system, some concurrent objectives have to be considered.

In this paper, we provide a review of the most popular approaches for SDST-MOHFSPs. Although the conventional approaches to solve scheduling problem use 'exact methods'; they are still incapable of solving real-world applications. This hindrance becomes more severs in the case of optimizing SDST-MOHFSP. Therefore, the study of SDST-MOHFSP has attracted a great many researchers' attention to develop effective and proficient approaches. Apart from this review, first a detailed theoretical discussion comprising a technical background of flowshop and setup time concepts is presented. A thorough discussion of terms and technical aspects of flowshop and the setup time concepts as well as the required references for further follow-up are also provided. In this section a new generic mathematical programming model for a general version of SDST-MOHFSP is proposed. Based on the review of the scheduling literature, the proposed model has not been considered in previous papers.

2. Theoretical Discussion

In a pure flowshop problem, there is a shop containing *m* different machines, and each job consists of *m* operations, each of which requires a different machine. The ith operation of the job must be executed on the ith machine. No machine can perform more than one operation simultaneously. For each operation of each job, execution time is specified. The machines can thus be numbered 1, 2, …, *m* and the operations of job *k* can be numbered (1, *k*), (2, *k*), …, (*m, k*). In the general case, some jobs may require fewer than *m* operations (the missing operations are considered as operations with the zero-processing times), but the precedence structure of all the operations are exactly similar to the pure flowshop.

A Hybrid Flowshop Scheduling (HFS) consists of series of production stages (at least two stages), each of which has several machines, (identical, uniform and unrelated) operating in parallel. Some stages may have only one machine, but at least one stage must have multiple machines. The flow of jobs trough the shop is unidirectional, that is; all jobs are processed following the same production flow: Stage 1, stage 2, . . ., stage *m*. A job might skip any number of stages provided it is processed in at least one of them. The general structure of HSF can be depicted as Figure 1.

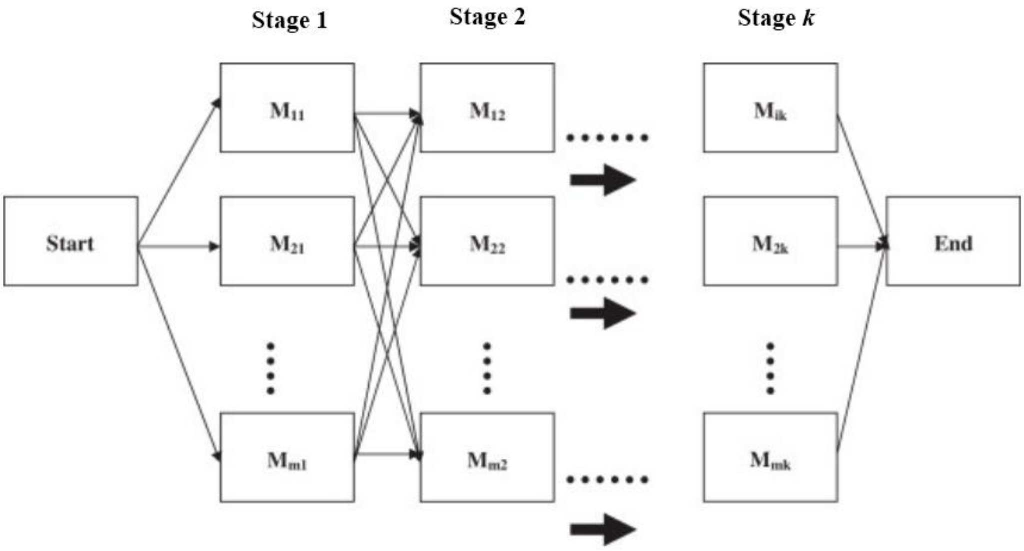

Figure 1. A Schematic of Hybrid Flowshop Model.

Concerning the machine setup, there are two distinct cases; sequence-dependent setup times and sequence-independent setup times. Where the setup times for the jobs are independent of job sequence, they can be included in processing times. The terms 'setup' and 'changeover' are occasionally used interchangeably. Operations carried out at each machine, which are not directly related with the processing of the jobs, are generally referred to as setup times [51].

2.1 Abbreviations

For reader convenience, we use abbreviation for terms that are frequently repeated in this paper.

Table 1. Abbreviation of Terms.

Abbreviation	Description	Abbreviation	Description
ACO	Ant colony optimization	IA	Immune algorithm
AEC	Augmented ε-constraint	ICA	Imperialist competitive algorithm
AVN	Adaptive variable neighborhood	IG	Iterated greedy
BIS	Biological immune system	ILS	Iterated local search
BO	Bacteria optimization	IPG	Iterated Pareto greedy
BS	Beam search	IWO	Invasive weed optimization
CCA	Colonial competitive algorithm	MA	Memetic algorithm
CHA	Constructive heuristic algorithm	MBO	Migrating birds optimization
CS	Cuckoo Search	NN	Neural Network

Abbreviation	Description	Abbreviation	Description
DEA	Differential evolution algorithm	PEA	Pareto evolutionary algorithm
DGW	Discrete grey wolf	PSO	Particle swarm optimization
EA	Electromagnetism algorithm	RKGA	Random key genetic algorithm
FFO	Fruit fly optimization	SA	Simulated annealing
FLA	Frog-leaping algorithm	SM	Simulation model
GA	Genetic algorithm	SI	Swarm intelligent
GH	Greedy heuristic	TS	Tabu search
GIM	Genetic immune strategy	VNS	Variable neighborhood search
GRASP	Greedy randomized adaptive search procedure	WFA	Water flow algorithm

2.2 FSP (Solution Approach)

The FSP is always carried out in order to satisfy a criterion or set of a performance criterion that characterizes the nature of the FSP. In the following, we propose a new mathematical programming model for the identical parallel-machine SDST-MOHFSP, which can be an alternative to the existing models. First, we introduce the parameters and variables of this mathematical model.

Parameters

K_s: Number of machines in stage s.

M: A very large number.

N: Number of jobs.

P_{iks}: Operation processing time of job i on machine k at stage $i \epsilon N, k \epsilon Ks$.

S: Number of stages.

S_{ij}: Setup time of a machine where job i immediately precedes job j, $i, j \epsilon N$.

Variables

x_{isk}: The starting time of job i on machine k at stage s, $i \epsilon N, k \epsilon Ks$.

y_{ij}: A zero-one variable, takes 0, if job i immediately precedes j and takes 1 otherwise, $i, j \epsilon N$.

z_{isk}: A zero-one variable to verify if job i is on machine k at stage s, $i \epsilon N, k \epsilon Ks, s \epsilon S$.

$$Optimize f\{\tilde{x}/\tilde{x}=(x_{ij})\, i\neq j,\ i,j=1,2,\cdots,n\}=\{f_1(\tilde{x}),f_2(\tilde{x}),\cdots,f_l(\tilde{x})\} \tag{1}$$

Subject to:

$$x_{i(s-1)k}+p_{i(s-1)k}\leq x_{ist} \qquad \forall 1\leq i\leq N, 1\leq s\leq S,\ 1\leq k,t\leq K_s \tag{2}$$

$$-My_{ij}+x_{isk}+p_{isk}+s_{ij}\leq x_{jst} \qquad \forall 1\leq i,j\leq N, 1\leq s\leq S,\ 1\leq k,t\leq K_s, i\neq j \tag{3}$$

$$-M(1-y_{ij})+x_{jsk}+p_{jsk}+s_{ji}\leq x_{ist} \qquad \forall 1\leq i,j\leq N, 1\leq s\leq S,\ 1\leq k,t\leq K_s, i\neq j \tag{4}$$

$$t^s_{isk}=x_{isk} \qquad \forall\ 1\leq i\leq N, 1\leq s\leq S, 1\leq k\leq K_s \tag{5}$$

$$t^f_{isk}=x_{isk}+p_{isk} \qquad \forall\ 1\leq i\leq N, 1\leq s\leq S, 1\leq k\leq K_s \tag{6}$$

$$\sum_{t^s_{kir}}^{t^f_{kir}}\sum_{j=1, j\neq i}^{N}x_{isk}=0 \qquad \forall k=1,2,\cdots,K_s, s=1,2,\cdots,S \tag{7}$$

$$\sum_{k=1}^{K_s}z_{isk}=1 \qquad \forall 1\leq i\leq N, 1\leq s\leq S \tag{8}$$

$$x_{isk}\leq M\, z_{isk} \qquad \forall\ 1\leq i\leq N, 1\leq s\leq S, 1\leq k\leq K_s \tag{9}$$

$$x_{isk}\geq 0 \qquad \forall\ 1\leq i\leq N, 1\leq s\leq S, 1\leq k\leq K_s \tag{10}$$

$$y_{ij}=0\,\text{or}\,1 \qquad \forall 1\leq i,,j\leq n, i\neq j \tag{11}$$

$$z_{isk}=0\,\text{or}\,1 \qquad \forall\ 1\leq i\leq N, 1\leq s\leq S, 1\leq k\leq K_s \tag{12}$$

Eq. (1) presents the objective function relation. Inequality (2) insures that the succeeding operation of a job won't be started before the completion of its preceding operation. Inequalities of (2) and (3) simultaneously act to find either job *i* precedes job *j* or vice versa. That is; if the auxiliary binary variable y_{ij} takes 0, it forces the operation *k* of job *i* precedes the operation *k* of job *j*, and if y_{ij} takes 1, it forces the operation *k* of job *j* precedes the operation *k* of job *i*. At the same time these inequalities insure that the succeeding job on machine *k* to be started after completion of its preceding job and machine *k* setup time. Inequality (4) and Inequality (5) respectively define the starting time and fishing time of a job on a machine at any stage. These variables are then used in Eq. (7) to insure that only one job is assigned to a machine simultaneously. Eq. (8), insures that a job operation at any stage is assigned to no more than one machine. Eq. (9) implies that a job on any machine at any stage can get a positive value if and only if is associated zero-one variable takes 1. The other three relations (10) through (12), specify the type of variables.

The conventional approaches to solve the MOHFSP can be generally categorized as the exact and heuristic methods. Table 2 depicts a hierarchical division for the solution methods of

OHFSPs. There is a general consensus that multi-objective optimization methods can be broadly decomposed into two categories: Scalarization approaches and Pareto approaches [52]. While different names are used for these categories, the essential discriminator is the same. In the first group, the multi-objectives are weighted into a single objective or a series of single objective scalar problems. The formation of the aggregate objective function requires that the preferences or weights between objectives are assigned a priori, i.e. before the results. The Pareto methods, on the other hand, keep the elements of the objective vector separate throughout the optimization process and typically use the concept of dominance to distinguish between inferior and non-inferior solutions.

In other decision-making process view, the methods to solve multi-objective FSP are classified as: A priori methods, a posteriori methods, and interactive methods [42]. The a priori method is the simplest method enabling the decision maker to intervene before the decision process. This group of techniques includes the approaches in which either some achievable goals or a certain pre-ordering of the objectives can be performed by the decision maker prior to the research. These approaches primarily include lexicographic ordering, Ɛ-constraint, linear fitness combination, and nonlinear fitness combination methods [63].

In a posterior methods, a posteriori knowledge or justification is dependent on practice or empirical evidence. Actually, a priori justification is a certain kind of justification often contrasted with posteriori, justification. Review of literature reveals that a posteriori method in solving the multi-objective FSP are much ampler than a priori methods. The ‘posteriori’ multi-objective approach is more complicated since in this case, there is no single optimum solution, but rather lots of ‘optimum’ solutions [181]. A posteriori method aims at producing all the Pareto optimal solutions or a representative subset of the Pareto optimal solutions. A solution is called Pareto optimal objective values [153, 154].

2.3 Classification of Flowshop Models

The FSP can be clustered per different assumptions such as assumptions concerning machines; assumptions concerning job; assumptions concerning operating policies [84]. Based on these assumptions, different types of flowshop model can be defined. Blocking flowshop: In this model, there is no buffer storage between some or all the consecutive two stages. In this case if the downstream machine is not available the job stays on the current machine until the machine in the next stage becomes available [195]. Common due date flowshop: In this model, an exclusive due date is assigned to all jobs. Either the due date is given in advance, or the due date is fixed [183]. Distributed flowshop: This type of flowshop is an extension of the classical flowshop setting where more than one factory, each with machines disposed in series, is available to process jobs. This model connects two sub-problems; assigning jobs among factories and scheduling the assigned jobs within each factory [121].

Table 2. A Hierarchical Presentation of the Solution Methods for the FSPs.

	Optimization methods													
Exact	Heuristics													
	Meta heuristics							Heuristics						
	Population					Biological		Constructive			Improving			
	Evolutionary			Swarm										
	Differential	Mimetic	Genetic Algorithm	Particle Swarm	Ant (bee) Colony	Immune System	Neural Networks	Insertion methods	Iterated Greedy	Dispatching Methods	Neighborhood Search	Tabu Search	Local Search	Simulated Annealing

Generalized tardiness flowshop: In flowshop models, dealing with the tardiness criteria, there is a given due date for completion of the last operation of each job. In real world problems, however there is a considerable situation in which there is a given due date for each intermediate operation of a job. Considering minimization of tardiness, this model is called 'generalized tardiness flowshop' [69, 70, 74, 151]. Hybrid flowshop: In this model, a set of jobs has to be processed in a series of processing centers generally called stages, with a direct precedence structure, in which more than one machine operates in one or more of these stages [143]

No-wait or continues flowshop: In this model the jobs on machines has to be processed without interruptions between consecutive machines [199]. Proportionate flowshop: In this model, the processing time of any job at any machine is proportionate to the processing time on the first machine [206]. Order flowshop: In this model, the processing order of jobs on machines is the same for each subsequent step of processing [44]. Reentrant flowshop: By the standard assumption of the FSP, each job visits each machine exactly once. However, in the reentrant flowshop one or more jobs may need to be processed repeatedly at one or more stations [43]. There are some other types of flowshop problems which have been less renowned problems in the related literatures. Among those are, flowshop with deteriorating jobs [64]; limited buffer flowshop [223]; flow shop scheduling with resource-dependent processing times, etc. [23].

3. Review of the Previous Research Works

In this section, we review the most popular literature on SDST-MOHFSP optimization. This section is divided into seven subsections. Although the review of publications is devoted to the SDST-MOHFSP, however since the majority of the multi-objective cases are somehow the extended version of the single objective problems, it would be valuable to review them as well.

3.1 SDST-MOHFSP (Exact Methods)

The SDST-FSPs, have been proved to be NP hard [84, 149]. Considering the fact that the HFSP is an extension of the regular FSPs, with more than one machine per stage, to find the final schedule the assignment of jobs to machines and the sequence of jobs on each machine have to be decided concurrently. This makes the problem more intractable even for makespan [65].

Review of the related literatures indicates that exact solution approaches for the SDST-MOHFSP problem mostly rely on problem-specific Branch & Bound (B&B) algorithms. Due to the complexity of the problem, most of the efforts have focused on improving the objective functions bounds and discovering a more efficient lower and upper bound [97]. Moreover, B&B algorithms generally appear to be limited to situations where the objective is to minimize makespan or mean flow time. Although B&B is the ideal technique for obtaining the optimal solution of the SDST-MOHFSP, but its state of art is still waiting for a unified solution algorithm. Based on review of the related literatures, it appears that the attempts in presenting a more efferent mathematical program of the SDST-MOHFSP have been ignored. With this respect, in the technical discussion of this paper (Section 2.2), we proposed a more efficient MILP model for the general version of the SDST-MOHFSP with the parallel identical machines.

Although the SDST-MOHFSP has been considerably studied in the literature, a very limited has considered the exact solution approaches [33, 215, 219, 220, 237]. Maboudian and Shafaei [144] have addressed a two-stage SDST-MOHFFSP assembly flow shop scheduling problem with the objectives of minimizing makespan and maximum tardiness. First, the problem was formulated by a NMIP and a lower bound for the problem was obtained. Ramezanian et al. [191] studied a SDST-MOHFSP with bypass consideration for minimizing the sum of earliness and tardiness costs. A mathematical model was constructed in which several constraints such as the due dates of jobs, the job ready times, the earliness and the tardiness cost of jobs were involved. An exact solution as well as an adapted GA based on bypass consideration was proposed to solve the problem. Application of the hybrid flexible flowshop model in a real-world case problem was presented in [112, 113]. In this paper, the model with the STSD constraint was adapted in a paint company to minimize the makespan and a MILP was constructed. The problem SDST-HFFSP with time lags between every two consecutive stages was considered in [100]. In this work an exact method has been developed.

Xiao and Zheng [231] and [99] presented a continuous-time MILP model for the short-term batch scheduling with sequence-dependent setups in semiconductor assembly and test manufacturing to minimize the total completion time of the products. The decisions of allocating jobs to parallel machines at each stage and sequencing the jobs in a SDST-MOHFSP with the job limited waiting time between two consecutive stages to minimize sum of the linear earliness and quadratic tardiness has been considered in [21]. Gicquel et al. [75] employed a SDST-MOHFSP to model a real-world scheduling problem arising from a bio-process industry. An exact solution approach based on a discrete time representation and a MILP was proposed.

Wang et al. [229] applied a scheduling problem in a semiconductor manufacturing process to minimize the number of tardy jobs and makespan with sequence-dependent setup time, release time, due dates and tool constraints. A MIP model is proposed in which minimization of the weighted sum of makespan and heavily penalized tardy jobs tardy jobs are considered as the objective function and tardy jobs is adapted as the soft constraints. A SDST-MOHFSP with unrelated parallel machines to minimize the makespan, total tardiness and total earliness of jobs proposed in [114] and aa MIP was developed. Hecker et al. [93] presented some evolutionary algorithms while Hecker et al. [94] presented a modified GA algorithm, an ACO algorithm, and random search procedures for the problem with respect to two performance measures of Cmax and total machine idle time. Hecker et al. [94] showed that the modified GA algorithm improves the Cmax performance of the current practice by 9% while it improves the total machine idle time by 23%. Panahi [177] addressed a two-stage SDST-MOHFFSP, with parallel identical machines, and devised four different mathematical models to minimize the makespan and tardiness of jobs.

Two works have found considering ‘the learning effect’[142] and ‘forgetting effect’ on machine setup. One is the work of Tyagi et al. [226] who considered learning and forgetting effect of workers for SDST-MOHFFSP to minimize the weighted sum of maximum completion time and maximum tardiness. And the other is the study conducted in [204].

Hosseini [98] studied the problem of assembly SDST-HFFSP considering aging effects, and preventive maintenance to minimize the makespan. A mathematical modeling was presented and its validation was shown. Lee and Bang [128] considered a two-stage assembly-type SDST-HFSP with the objective of minimizing the total tardiness. To obtain an initial upper bound, a two-stage heuristic algorithm was used. A multi-product multi-period capacitated SDST-HFFSP has been presented in [192]. In this paper, a mathematical model was developed and compared with former models through which the superiority of the proposed model was proved.

3.2 SDST-MOHFSP (Heuristic Methods)

Heuristic algorithms are generally classified into, constructive and improvement heuristics. Constructive heuristics build a schedule from scratch by searching through the list of unscheduled jobs [73, 89, 102, 173, 228]. At each iterative step one or more jobs are selected and added to the schedule. The most classical and well-known method to build a schedule is a variant of the dispatching rules. Contrary to constructive heuristics, improvement heuristics start with an existing solution and apply some rules to improve the initial schedule [228]. Neighboring and local search, simulated annealing (SA) and Tabu search (TS) are the most popular procedure for improving the initial schedule [172]. In this section the existing research works adapting heuristic algorithms for SDST-MOHFSP will be presented separately per hybrid models, flexible models and assembly models.

One of the early attempts considering SDST-HFFSP is the work of Liu and Chang [136] in which a Lagrangian relaxation-based approach was presented. The proposed problem first formulated as a separable IP with synchronization constraints between production and machine usage. A SDST-HFFSP with the objective of minimizing the sum of setup times and costs presented in [123]. A Lagrangian relaxation was utilized, and an iterative heuristic algorithm was proposed. Kurz and Askin [124] presented three heuristic algorithms based on the greedy methods, insertion routines and the adaptation of Johnson's rule for a SDST/HFSP. Later, the same authors, Kurz and Askin [124] have examined scheduling in SDST-flexible flow lines to minimize makespan [4]. Several heuristics, based on greedy methods, flow line methods, the insertion heuristic were developed.

Lin and Liao [133] addressed a two-stage SDST-HFSP in which the first stage includes SDST and the second stage comprises dedicated parallel machines. The problem was handled by proposing a constructive heuristic algorithm. Chang et al. [39] developed a setup time-based heuristic method to generate the initial solutions for the GAs. Later, Chang et al. [40] adapted a two-stage no-wait SDST-HFSP with several identical parallel machines to minimize the makespan of jobs. Two heuristic algorithms; one for sequencing and the other for assigning of jobs were proposed.

The problem of multi-stage limited buffer-SDST-HFFSP to minimize makespan was considered in [48]. Several constructive heuristics were developed and by the use of a SA. A research considering the extended version of the single objective SDST-MOHFFSP was conducted by Jungwattanakit et al. [106, 107]. Minimizing the positively weighted convex sum of makespan and the number of tardy jobs were defined as the objectives. Two constructive heuristics and three improvement algorithms were contemplated. Later, the same authors employed the results of this research and applied them to a textile industry problem [108].

Naderi et al. [169] have used SA to handle a SDST-MOHFSP with transportation times to minimize the total completion time and total tardiness. The efficiency and effectiveness of the proposed SA were inferred from all the computational results. Naderi et al. [168] explored a SDST-HFFSP and developed an IG to minimize the total weighted completion time of jobs. Another work presenting SA for a SDST-HFSP is found in [217]. The results of applying SA to the problem were compared with RKGA and results revealed the superiority and effectiveness of the SA.

A GRASP for solving a hybrid SDST-MOHFSP to minimize the maximum tardiness and the sum of completion times was presented in [50]. Abiri et al. [1] and the author of [34] proposed TS for the scheduling of a SDST-HFSP. In this algorithm a selection mechanism as well as *n*-dimensional search mechanism was incorporated to assist the search for a near optimal solution. Li et al. [129] acquired the seamless tube plant of Baoshan Iron and Steel Complex production process and adapted a SDST-MOHFSP. The problem was first formulated by a MONLMIP model and a two-stage heuristic algorithm was proposed. A modified GA and a LS were combined to develop this heuristic.

Kia et al. [117] presented some scheduling rules to sequence jobs in a dynamic SDST-MOHFFSP considering two distinct measures of performance, the mean flow time and the mean tardiness. A discrete-event simulation model and eight heuristics were adapted. Hatami et al. [90] and [2] considered a bi-objective three-stage assembly SDST-MOHFSP to minimize the mean flow time and maximum tardiness. A mathematical model with a new lower bound was proposed. Additionally, two meta-heuristics; a SA and a TS were proposed. Naderi et al. [171] considered the SDST-HFFSP to minimize makespan. A dynamic dispatching rule heuristic and an ILS metaheuristic were proposed. Eren [56] adapted a bi-criteria SDST-HFSP and proposed three improvement heuristics similar to NEH procedure to minimize weighted sum of total completion time and makespan criteria. In work of Mirsanei et al. [156] the problem of SDST-HFSP with parallel identical machines to minimize the makespan was studied and a SA was developed.

Mousavi et al. [160] considered the problem of scheduling of a SDST-MOHFSP to minimize the makespan and total tardiness. A bi-objective heuristic with the neighborhood structure was proposed for searching Pareto-optimal frontier. Seyedi et al. [205] adapted a similar model with blocking times between each stage to minimize the weighted mean completion time and makespan and a SA and a TS were proposed. Another work considering a similar model to minimize the weighted mean completion time and makespan were presented in Maleki-Darounkolaei et al. [146]. In this paper a meta-heuristic method based on SA was constructed.

Gupta and Sharma [84] introduced a SDST-MOHFSP to minimize the rental cost of machines with minimum makespan. Johnson algorithm [102] was used and a simple constructive heuristic algorithm was proposed. Wang et al. [229] investigated a SDST-MOHFSP with release time, due dates and tool constraints in a semiconductor manufacturing process to minimize the number of tardy jobs and makespan. An efficient dispatching rule and some LS heuristics were proposed. Ciavotta et al. [45] considered a permutation SDST-MOHFSP and presented a simple algorithm named as restarted iterated Pareto greedy minimizing the makespan and the total flow time. The efficiency of the GRASP for solving the SDST-HFSP with the controllable processing times and assignable due dates was evaluated in [12]. For handling this problem, a SA approach was used h and a non-regular optimization criterion based on due dates was considered. Lee et al. [127] considered a two-stage SDST-HFSP which was initiated from a real-world case of job sequencing for ‘Automated Optical Inspection Machine (AOIM)’. A method combining two heuristics has been developed.

The majority of the works dealing with SDST-MHOHFSP have considered permutation scheduling problems. A work dealing with non-permutation flowshop problem is introduced in [210]. In this paper, a MILP is developed to solve non-permutation SDST-HFSP to minimize makespan, increasing service level, and maximizing job priorities. This model is actually the generalization of model proposed by Mehravaran and Logendran [150]. Another non-permutation FSP was found in [10]. This work addressed a general three-objective non-permutation SDST-MOHFSP to minimize the makespan, the sum of flow time and maximum tardiness simultaneously. The problem was formulated as a MILP and a heuristic algorithm based on the ε-constraint method was proposed.

A study dealing with a SDST-HFSP problem where jobs are organized in families according to their machine settings and tools has been presented in [142]. The problem is complicated by the considerations of family setup time in two situations: It varies with the job families in different stages and it changes when a machine is released from processing one job family to another. This problem with a non-delay scheduling heuristic in which no machine is kept idle has been previously cited in several literatures [31, 66]. In this work, it was shown that the inserting idle time into a non-delay schedule can further reduce the total setup time as well as makespan. Hosseini [98] studied the problem of assembly SDST-HFSP considering aging effects, and preventive maintenance to minimize the makespan. A mathematical model and four heuristic algorithms, based on the Johnson's algorithm, were proposed. Harbaoui et al. [88] applied a SDST-HFSP with dedicated machines, and time lags configuration in the pasta production industry. They considered the objective of minimizing the makespan of all jobs and proposed a mathematical model. The proposed model was used to obtain an upper and a lower bound and applied to a GA.

Du et al. [54] presented a distributed assembly flowshop scheduling problem with four realistic extensions, i.e. random nature of processing times, stochastic SDST, stochastic job releases times, and the no-wait constraint. A method by integrating PSO-based exploration, SA-based LS was proposed to minimize the stochastic makespan of the jobs.

The (ILS) has been widely used to tackle a variety of single scheduling optimization problems [105] due to its powerful search ability. However, applying ILS to solve multi-objective scheduling optimization problems is scanty. Xu et al. [234] and [86] designed a multi-objective ILS (MOILS) for the SDST-MOHFSP to minimize the makespan and total weighted tardiness of all jobs. A Pareto-based variable depth search was adapted in the in the multi-objective LS phase of the MOILS. The search depth was dynamically adjusted during the search process of the MOILS to strike a balance between exploration and exploitation. Pan et al. [176] proposed nine heuristic algorithms to minimize the makespan for the SDST-HFSP. The first six algorithms were trajectory-based metaheuristics, including three variants of ILS and three variants of IG. The remaining three algorithms were population-based metaheuristics, namely, the improved FFO, the improved MBO, and the discrete ABC optimization. A path-relinking-based cooperative search, a diversity control scheme, and a diversified initialization approach were also constructed.

In the SDST environment, the operational cost of machines is related to the setup time required for processing of jobs on different machines. Sharma et al. [208] considered a SDST-MOHFSP to minimize the operational cost and the makespan subject to minimum total elapsed time of jobs, where the operational cost of machines is related to the setup time. A heuristic algorithm was proposed to find the latest time so as to reduce operating time and hence the operational cost with minimum makespan. A research study for extending the work of Gupta et al. [84] was conducted by Sharma et al. [209]. In this paper, the concept of job-block in SDST-MOHFSP to minimize total completion time and the machine rental cost is considered. A constructive heuristic, based on Johnson procedure was developed to find optimal or near optimal sequence of jobs processing.

A multi-product multi-period capacitated SDST-HFFSP has been presented in [192]. In this paper, three mixed-integer programming-based heuristics, all based on iterative resolutions of reduced-size MIPs and rolling horizon were implemented to solve the model.

3.3 SDST-MOHFSP (Evolutionary Approaches)

Evolutionary Algorithms (EA) are stochastic search and optimization heuristics originated from the classic evolution theory and consist of several heuristics, which are able to solve optimization tasks by imitating some aspects of natural evolution [152]. Among the set of search techniques, the development of EA has been very important in the past two decades [76]. Most of the present implementations of EA come from any of these three basic categories: GAs, evolutionary programming and evolutionary strategies. EA use mechanisms inspired by biological evolution, such as reproduction, mutation, recombination, and selection. Candidate solutions and the fitness function regulates by the quality of the solutions. Evolution of the population then takes place after the repeated application of the above operators. Due to their random nature, EA are never guaranteed to find an optimal solution for any problem, but they will often find a good solution if one exists [5, 47, 71].

Early work considering a SDST-HFSP to minimize makespan is presented in [101]. In this work, a scheduling problem in a PCB assembly line for inserting electronic components to produce a variety of PCB types was investigated ans a GA was presented. Among other early attempts applying EA for the SDST-MOHFSP is the work of Cochran et al. [46]. In this paper, a two-stage MPGA with objectives of makespan and total weighted tardiness was proposed. Lee and Asllani [126] propose two alternative solutions; a MILP, and genetic programming for solving flexible SDST-MHOHFSP. Three criteria of minimizing the setup times, number of tardy jobs, and makespan were considered to define the objective function.

Ponnambalam et al. [184] presented a multi-objective evolutionary search algorithm using a TSP and GA for SDST-MOHFSP to minimize makespan, mean flow time and machine idle time. Kurz and Askin [124] examined scheduling in SDST-HFFSP to minimize makespan. Several heuristics were developed, based on greedy methods, flow line methods, the insertion heuristic for the TSP and the RKGA. A multi-objective evolutionary search algorithm using a travelling salesman algorithm and the GA for a SDST-HFFSP was proposed in [184]. Jungwattanakit et al. [107] considered a SDST-MOHFFSP minimizing the positively weighted convex sum of makespan and the number of tardy jobs. In this paper, two constructive heuristics were proposed. Ruiz and Maroto [196] provided a GA, for a multi-stage SDST-HFSP, with unrelated parallel machines at each stage, to minimize the makespan. A paper dealing with the solution of a SDST-MOHFSP with unrelated parallel machines to minimize the makespan and the number of tardy jobs was presented in [107]. A constructive heuristic such as the algorithms given by Palmer [178]; Campbell et al. [36]; Gupta [81, 82]; and Dannenbring [49] an insertion heuristic has been proposed. Furthermore, some heuristics such as a GA and a SA approach were presented.

Jungwattanakit et al. [108] considered a SDST-MOHFFSP with parallel machines and applied it to a real-world textile manufacturing scheduling problem. Minimizes a convex combination of makespan and the number of tardy jobs were contemplated and a MIP was presented. A GA and a constructive dispatching rule were also explored. Hendizadeh et al. [96] considered a SDST-MOHFSP and presented a MOGA to minimize the makespan and the total flow time. Chang et al. [41] designed an artificial chromosome generating mechanism for the SDST-MOHFSP to minimize the makespan and the maximum tardiness. It was shown that embedding artificial chromosomes into NSGA-II and GA could speed up the convergence of GA and could improve the solution quality as well.

Qian et al. [185] proposed an effective hybrid algorithm based on DE to solve SDST-MOHFSP with limited buffers. Mansouri et al. [148] addressed a two-machine SDST-MOHFSP to minimize the setup times and the makespan. Two multi-objective metaheuristics based on GA and SA were proposed. Naderi et al. [169] proposed SA to handle a SDST-MOHFSP with the objectives of minimizing the total completion time and total tardiness. A group scheduling problem in an SDST-HFFSP to minimize makespan was investigated in [241]. Two metaheuristic algorithms, based on an SA and a GA, have been proposed. Several constraints, such as SDST, machine release date, and time lags were considered in a hybrid flexible flowshop model, to adapt it for the real-world problems, and a GA was proposed in [242]. The performance of the proposed algorithm was then compared with three literature cited methods of SPT, LPT and NEH and the results are reported. Alfieri [9] studied a real - life SDST-MOPFSP in a cardboard company which differs from the conventional flowshop problem in several characteristics, such as multi-machine stations, work calendars on resources, reentrant flows, external operations, and transfer batches between stations. Maximum tardiness, total weighted tardiness and total setup times were defined as the objective functions. Another paper considering the stochastic machine breakdown with the mean makespan criterion is the work of Gholami et al. [77]. In this work, the process of incorporate a digital computer simulation into GA was describe.

Yaurima et al. [235] adapted a GA for a SDST-HFSP with unrelated machines, availability constraints, and limited buffers. The proposed GA was a modified version of the algorithm for a problem without limited buffers in which additional limited buffer constraints and use of a new crossover operator and stopping criteria were contemplated. Application of a two-stage SDST-HFSP in metal working operations was investigated in [141]. Several assumptions such as; blocking between two stages with no intermediate buffer storage, unavailability of machines in some designated periods due to the preventive maintenance and machine breakdown were considered. A GA was proposed to minimize the makespan of the jobs. Kumar and Dhingra [122] presented four modified heuristics for preliminary viable SDST-MOFHSP with minimizing the weighted sum of total weighted tardiness and makespan. Another application of the hybrid flexible flowshop model in a real-world case problem is presented in [113]. In this paper, the model with the SDST constraint was adapted in a paint company to minimize the makespan. To solve the problem, an MILP was constructed and two heuristic methods were employed.

The problem of SDST-HFFSP with jobs precedence structure was considered in [227]. This study is a continuation of Ruiz et al [197]in which in addition to the precedence constraint several other characteristics like overlaps, machine release dates, time lags is contemplated. Karimi et al. [111] conducted a study to minimize the SDST-MOHFFSP by a multi-phase algorithmic approach. A MPGA to search Pareto optimal solution of a SDST-MOHFSP to minimize the total weighted tardiness and the maximum completion time was presented in [243]. The proposed MPGA comprises two stages. First stage applies combined objective of mentioned objectives and second stage uses previous stage's results as an initial solution. Khalili and Tavakkoli-Moghaddam [115] performed a study dealing with SDST-MOHFSP with the transportation times between machines, to minimize the makespan and total weighted tardiness. In this work, a MOEM was presented. A SDST-MOHFSP with limited buffers to minimize the makespan or the total tardiness of jobs was considered and some heuristic and metaheuristic approaches in the form of GAs were proposed in [87]. The resolution of several specific instances from the literature with the adaptations of NSGA-II and sub-population GA suggested that the proposed algorithms were effective and useful methods for solving this problem.

Lin and Ying [134] contemplated a manufacturing cell in the form of SDST-MHOFSP and present a two-level multi-start simulated annealing (TLMSA) heuristic with the objectives of simultaneously minimizing the makespan and the total flow time (or total tardiness). Gómez-Gasquet et al. [79] developed a GA to minimize the makespan of the schedules in a SDST-HFSP. The job preemption may improve the schedule performance in case of the HFSPs. A study dealing with a SDST-MOHFSP, with the objectives of minimizing the weighted sum of completion time and maximum tardiness was presented in [203]. A GA in which its parameters are tuned with a RSM has been proposed. Acosta et al. [3] addressed a SDST-MOHFFSP and proposed a GA to minimize makespan and the due dates related objectives.

Fadaei and Zandieh [58] have studied a SDST-MOHFSP to minimize the makespan and the total tardiness. In this study, three multi-objective algorithms, MOGA, sub-population GA-II and NSGA-II were proposed. A two-stage assembly SDST-HFSP with the objective of minimizing the makespan of all jobs was studied in [60]. A mathematical model was presented, and a series of heuristic algorithms based on the basic idea of Johnson algorithm is proposed. To further this work, Fattahi et al. [61] considered the same problem and proposed four algorithms. Due to the existence of sequence dependent setup times, the parts of the same kind were grouped in a block. This transformation then led to a three-decision making process. A heuristic was used for determining the number of blocks, Johnson's and NEH algorithm was employed for sequencing the parts and finally GA and SA were adapted for sizing the blocks. This work motivated the development of a branch and bound approach for the problem [62]. Maleki-Daronkolaei and Seyedi [147] studied a three-stage MOHAFSP with SDST to minimize the weighted sum of mean completion time and the makespan of jobs. Two meta-heuristic algorithms in the form of the VNS and SA were developed. To enhance the performance of the SA, the Taguchi method was adapted.

The scheduling literature revealed that limited works considering the time lag in the SDST-MOHFSPs have been presented. A paper dealing with the time lags and SDST in a HFSP is addressed in [59]. In this paper, first a MILP to minimize the makespan of schedule was proposed. Then, a GA and three heuristic algorithms were proposed. Saluja and Jain [198] presented optimization of makespan for a multi-stage SDST-HFFSP using GA. A restart scheme as suggested by Ruiz and Maroto [196] has also been applied to prevent the premature convergence. The paper also assessed the effect of lot splitting size on makespan. Ebrahimi et al. [55] studied a SDST-MOHFSP to minimize the makespan and the total tardiness. Two metaheuristic algorithms, NSGA-II and MOGA, were presented. Experimental results indicated that the NSGAII outperforms MOGA.

A due date related criteria scheduling problem is tackled by Gholami and Rajaee Abyaneh [78]. In this paper, the SDST-MOHFSP with unrelated parallel machines at each stage to minimize the sum of earliness and tardiness was studied. A hybrid MA, and a PSO algorithm combined with genetic operators were introduced. The results of one-way ANOVA demonstrated that the proposed MA outperforms a literature cited SA. Sioud et al. [213] presented a GISMOO algorithm adaptation to solve a SDST-MHOFSP to minimize the makespan and the total tardiness. Mohammadi [159] developed an approach based on GA for solving the SDST-MOHFSP. Zandieh and Hashemi [244] studied the problem of group scheduling in a SDST-HFFSP with machine random breakdown and integrated simulation in GA to minimize the expected makespan of jobs.

A hybrid GA have been developed for solving a SDST-MOHFSP to minimize the total weighted squared earliness, makespan, and the number of tardy jobs in [53]. Mousavi and Zandieh [162] considered the problem of scheduling independent jobs in a SDST-MOHFSP to minimize convex combination of the makespan and the total tardiness. A procedure based on HSA-GA and LS was proposed. In a later attempt, Mousavi et al. [163] contemplated the same problem with re-entrant lines and position-dependent learning effects. A bi-objective of makespan and total tardiness were defined as the scheduling criteria. A solution method based on GA was proposed to solve the problem. Recently the same authors [163] proposed a VNS priority based for reentrant MOHFSP to minimize maximum makespan together with total tardiness objectives.

Li and Ma [131] proposed a MOMA, to solve a SDST-MPFSP with makespan and total flow time. Nejati et al. [175] considered a two-stage assembly SDST-HFSP to minimize the sum of weighted completion times of products in each shift in order to furnish better machine utilization for the next shifts. A GA and a SA were also developed. Lu et al. [138] performed a research for solving scheduling problem of a real-world manufacturing welding process. A multi-objective mathematical model was first constructed. Then, a MODGWO, considering not only production efficiency but also machine load on this real-world scheduling case was proposed.

Consideration of the transportation time and energy-efficient in a real-world SDST-MOHFSP was investigated by [139]. In this study, a multi-objective mathematical model considering both makespan and energy consumption was formulated. Then, a hybrid multi-objective backtracking search algorithm was proposed. In another work, the proposed model was employed for scheduling of a real-world welding process [140]. Three types of dynamic events, namely, machine breakdown, job with poor quality and job release delay to minimize makespan, machine load and instability simultaneously, were considered hybrid MOGWO was proposed.

To achieve excellence in manufacturing, goals like lean, economic and quality production with enhanced productivity play a crucial role. Satyanarayana and Pramiladevi [202] considered this environment in conjunction with three technical areas: variation reduction, equipment reliability and production scheduling. They proposed five heuristics based on HGA for a SDST-MOHFSP to minimize the makespan and the squared total tardiness. Sivapragasam and Suppiah [214] proposed several dispatching heuristics as well as a GA to minimize total weighted tardiness of a SDST-HFFSP with the identical parallel machines.

Review of the literature reveals that application of the simple priority rules for solving the flowshop scheduling problem has been widely studied [67, 68, 72]. Although these priority rules are not efficient enough due to simplicity and lack of general insight, composite dispatching rules have an appropriate performance because they result from experiments. A dynamic SDST-MOHFSP to minimize the mean flow time and the mean tardiness is studied in [118]. Four composite dispatching rules are proposed to solve the anticipated problem by applying genetic programming framework.

3.4 SDST-MOHFSP (Hybrid Metaheuristic Methods)

The performance of the metaheuristic approaches, for solving FSPs can be remarkably enhanced through the hybridization of metaheuristics with other optimization routines. In fact, choosing a suitable combination of complementary algorithmic concepts can be the key for achieving top performance in solving many hard optimization problems [29]. The hybridization of metaheuristics with heuristics is quite popular, especially for what concerns the use of LS methods inside population-based methods. In contrast to the early days of metaheuristic research, the last 5-10 years have produced a large number of algorithms that simply do not fit into a single metaheuristic category. This is because these untraditional approaches combine various algorithmic ideas, often originating from several branches of artificial intelligence [28]. Another hybrid in this category are hyper-heuristics. Hyper-heuristics comprise a set of approaches that are motivated by the goal of automating the design of heuristic methods to solve hard computational search problems. An underlying strategic research challenge is to develop more generally applicable search methodologies [35]. In this section, a review of works considering hybridize metaheuristic algorithms for solving SDST-MOHFSP will be described.

Behnamian et al [16] proposed a hybrid metaheuristic for the minimization of makespan in a SDST-HFSP with parallel machines. The proposed solution algorithm comprises three components; an ACO, a SA, and a VNS. It is shown that the hybridization of an ACO, SA with VNS, benefits the advantages of these three individual components. They [17] then adapted a SDST-MOHFSP with the objectives of minimizing the makespan and sum of the earliness and tardiness of jobs, and presented a multi-phase method. A research proposing the multi-phased method was conducted in [239]. In this paper, a hybrid GA is presented to minimize the makespan of jobs. The proposed GA is then compared with a literature cited

RKGA and superiority performance of the algorithm is concluded. Naderi et al. [170] considered a DST-MOHFSP to minimize makespan and maximum tardiness. A SA with 'Migration mechanism', and a new operator, called 'Giant leap' was proposed to bolster the competitive performance of SA through striking a compromise between the lengths of neighborhood search structures. The proposed SA was then hybridized with simple LS to promote the quality of the final solution.

Due-date assignment problems have gained considerable attention due to the industrial focus in the just-in-time concept. Behnamian et al. [18] discussed the problem of machine scheduling in which jobs should be completed at times as close as possible to their respective due dates. Both the earliness and the tardiness penalty were used as the criteria to obtain the optimal schedule. To achieve these objectives, three hybrid metaheuristic algorithms; a SA and VNS, an ACO and a VNS, an ACO, SA, and VNS were proposed. To further their previous study, Behnamian et al. [19] have focused on scheduling of a SDST-HFSP to minimize the sum of earliness and tardiness of jobs. A hybrid algorithm by combining VNS, ACO and SA was constructed. Afterward, the same authors [20] were considered a SDST-MOHFSP to minimize makespan and the total earliness/tardiness of the schedule and proposed a three-stage hybrid metaheuristic algorithm. Another due-date determination problem was considered in [23]. A SDST-MOHFSP in which jobs must be completed at times as close as possible to their respective due dates was contemplated. To achieve this goal, a bi-objective measure of performance in the form of minimizing sum of the earliness and tardiness of jobs was adapted. Three hybrid metaheuristics; one by combining SA and VNS, the second by composing ACO and VNS, and third by integration of ACO, SA and VNS, were developed.

A two-stage reentrant SDST-HFSP to minimize makespan, has been presented in [95]. Due to the intractability of the reentrant hybrid flow shop model [175], 13 heuristic algorithms, a HGA and a RKGA were proposed. Behnamian et al. [24] have addressed a SDST-HFSP with parallel identical machines to minimize the makespan and proposed a hybrid metaheuristic algorithm. This algorithm was constructed by integration of several features from ACO, SA and VNS. A SDST-HFFSP considering non-deterministic and dynamic arrival of jobs, to minimize the average tardiness of jobs, was studied in [119]. A dispatching rule and a hybrid GA have been developed and have incorporated in a discrete event simulation model. Jolai et al. [104] anticipated the problem of scheduling a set of jobs in a no-wait SDST-HFFSP to minimize the maximum completion time. Three metaheuristic algorithms in the forms of a population-based SA, an adapted ICA, and a HICA have been developed. A parallel machine scheduling problem considering lot streaming and setup time, both sequence dependent and sequence independent, for solving a case problem of a solar industry was modeled as a HFSP in Wang et al. [230]. In this industry, finding the optimal sub lot size, sequencing the sub lots, and assigning machines to the sub lots must be carried simultaneously. To perform this, a MILP model with consideration of SDST and SIST was developed and a hybrid-PSO was constructed to find its solution.

Behnamian and Zandieh [25] proposed a hybrid metaheuristic that hybridized the PSO, SA and VNS to solve a SDST-MOHFSP with position-dependent learning effects, and the consideration of tardiness together with earliness penalties as the measures of effectiveness. A multi-stage flowshop group scheduling with SDST and transportation times to minimize makespan was presented in [135]. A hybrid algorithm by combination of a PSO and a GA was constructed to find the sequence of jobs in each group and to sequence the groups simultaneously. Campos and Arroyo [38] focused on a three-stage assembly SDST-MOHFSP to simultaneously minimize the total flow time and the total tardiness of the jobs. In order to obtain near Pareto optimal solutions, a NSGA-II coupled with the IG was developed. Behnamian et al. [26] studied scheduling of a SDST-MOHFSP with makespan and sum of the earliness and tardiness as optimization criteria. In this paper, a hybrid algorithm by incorporating LP-metric method in PSO was proposed. The solution quality of the proposed algorithm was then improved by applying a PSO and a VNS. In another study, Behnamian [27] considered the simultaneous effects of learning and deterioration on a SDST-MOHFSP to minimize earliness, tardiness, makespan, and total worker employing costs. A hybrid metaheuristic based on CCA and VNS was proposed.

Sioud et al. [211] presented different LS algorithms to solve a realistic variant of the SDST-HFFSP with the objective of minimizing the total tardiness. The same authors, then suggested three new metaheuristic algorithms in the forms of a GA, an HGA and an ACO algorithms [212]. The algorithms were compared with ILS of Naderi et al. [171] and the GA of Ruiz and Maroto [196] and indicated that the proposed algorithms generate better solutions. A paper dealing with anticipatory machines setup on a SDST-HFFSP, with uniform parallel machines and no-wait between stages, to minimize maximum makespan was presented in [190]. In this paper, principles of meta-heuristic algorithms such as; IWO, VSN, and SA algorithms were hybridized as solution method.

Satyanarayana and Pramiladevi [201] formulated a SDST-MOHFSP to minimize the weighted sum of total weighted squared tardiness, the makespan, the total weighted squared earliness and the number of tardy jobs. Three heuristics based HGA, named SHGA1, SHGA2, and SHGA3 were proposed. Experiments were conducted on the benchmark problems of [224].

3.5 SDST-MOHFSP (Nature-Inspired Methods)

Nature teaches us how to solve complex problems efficiently and effectively. The main objectives of the nature inspired optimization are to develop computational models and algorithms inspired from natural intelligence found in physical, chemical, social and biological systems. Nature inspired optimization algorithms have been used to solve many difficult optimization problems, since their notable successes in solving optimization problems where other algorithms fail. Nature-inspired algorithms are the set of bio-inspired and swarm optimization algorithms which have innovative problem-solving methodology to provide good performance. Swarm intelligence and bio-inspired algorithms form an efficient and effective approach in the developments of new algorithms inspired by nature [179].

Bio-inspired algorithms are based on the structure and operational of complex natural systems that covers a wide variety of computational methods. They are a problem-solving methodology which has the ability to describe and resolve complex relationships from intrinsically very simple initial conditions and rule [32]. Swarm intelligence is the collective behavior of decentralized, self-organized systems, natural or artificial [14]. The concept is employed in work on artificial intelligence [30]. Collective Intelligence emerges through the cooperation of large numbers of homogeneous agents in the environment. With swarm intelligence, the developed algorithms need to be flexible to internal and external changes, to be robust when some individuals fail, to be decentralized and self-organized [15]. In this section, a review of works considering bio-inspired algorithms and swarm intelligence algorithms for solving SDST-MOHFSP will be described.

Zandieh et al. [238] addressed a SDST-HFSP to minimizing the schedule makespan. An IG is proposed and it is shown that the developed algorithm outperforms the GA of Kurz and Askin [125]. Tavakkoli-Moghaddam et al. [221] investigated a multi-objective model for a no-wait SDST-MOHFSP that minimizes both the weighted mean completion time and weighted mean tardiness. A hybrid algorithm based on the features of BIS and BO to find Pareto optimal solutions was proposed. A mixed-model assembly line sequencing problem is modeled by a SDST-MOHFSP to minimize the total utility work, total production rate variation, and total setup cost in [187]. A hybrid multi-objective algorithm based on shuffled FLA and BO was deployed.

In the majority of industries, a machine can be unavailable for many causes, such as unanticipated breakdowns, or due to the preventive maintenance. The unanticipated machine breakdowns are considered as the stochastic breakdown. The problem of stochastic machines breakdowns, to optimize objectives based of the expected makespan was studied in [240]. In this paper it was shown that the methods based on a colonial selection principle and an affinity maturation mechanism of the immune response were suitable approach of the addressed problem. Jolai et al. [103] considered a SDST-HFSP with machine availability constraints to minimize makespan. A metaheuristic based on MA incorporating some advanced features was present. Behnamian and Zandieh [21] proposed a discrete version of CCA to determine a schedule that minimizes sum of the linear earliness and quadratic tardiness in a SDST-MOHFSP with simultaneously considering effects of sequence-dependent setup times and limited waiting time. Tavakkoli-Moghaddam and Amin-Tahmasbi [223] presented a mathematical model for a SDST-MOHFSP considering minimization of the makespan and the weighted mean total earliness/tardiness. An MOIS was proposed to solve the anticipated problem

A SDST-HFFSP, with the time lags between every two consecutive stages, was considered in [100]. A meta-heuristic algorithm based on IA has been developed to minimize makespan. Jolai et al. [104] anticipated the problem of scheduling a set of jobs in a no-wait SDST-HFFSP to minimizing the maximum completion time. A hybrid algorithm based on SA, ICA, and HICA was constructed. Pargar and Zandieh [180] studied an automatic scheduling problem in which processing times do not differ according to repetition of job or process sequences.

In this research, a SDST-MOHFSP with learning effect for minimizing the weighted sum of makespan and the total tardiness is considered. A meta-heuristic approach called WFA which has the feature of multiple and dynamic numbers of solution agents is proposed.

A SDST-HFSO, in which jobs can be preempted, to minimize the weighted sum of makespan and maximum tardiness was presented in [194]. Three meta-heuristic algorithms based on a GA, an ICA, and an HGA were constructed. An artificial immune system for the SDST-MOHFSP is developed to minimize the makespan, tardiness, earliness and total completion time in [200]. The proposed approach was compared with the SA and B-GRASP approach. Computational experiments indicated that the proposed algorithm outperformed the SA and B-GRASP.

The problem of no-wait two-stage SDST-HFFSP considering unrelated parallel machines, and different ready times was studied in [186]. A hybrid algorithm based on ICA combined with SA, VSN and GA was presented. Attar et al. [13] adapted a SDST-MOHFSP with some useful constraints including of the limited waiting times between every two successive operations, unrelated parallel machines at least one stage. A MILP was provided to simultaneously minimize the total weighted tardiness and the maximum completion times. Then two metaheuristic methods based on the Pareto approach, multi-objective PSO and SPEA-II was developed. Sioud et al. [213] offered a hybrid GISMOO adaptation to solve a SDST-MOHFSP. The makespan and the total tardiness were defined as the two objectives.

A SDST-MOHFSP with unrelated parallel machines at each stage under minimization of the sum of earliness and tardiness has been studied in [189]. A hybrid MA-PSO algorithm combined with genetic operators was presented. The performance measure of the total tardiness of sequences for the assembly SDST-HFSP has been addressed in [7]. Six different algorithms; IA, GA, two versions of SA, and two versions of cloud theory-based SA were proposed. additionally, the proposed IA was combined with the rest of the algorithms resulting in a total of eleven algorithms. Khalili and Naderi [116] considered scheduling of no-wait SDST-MOHFFSP and proposed a bi-objective ICA to minimize both makespan and total tardiness of jobs. A research considering a multi-layer assembly SDST-HFSP with identical parallel machines to minimize the makespan was conducted in [120]. In this work, a nature inspired meta-heuristic method, CS algorithm was introduced. The problem of scheduling SDST-HFFSP with machine breakdown, machine eligibility, and machine-dependent setup time to minimizing total weighted earliness and tardiness was presented in [110]. Two metaheuristic algorithms namely a GA and ICA were proposed. Li and Ma [132] presented a MODABCA based decomposition, called MODABC/D, to solve a permutation SDST-MOHFSP with the objective to minimize the makespan and the total flow time. The performance of the proposed algorithms was tested on the Taillard [224] against the state of art algorithms and the results were reported.

Rahmanidoust et al. [188] proposed three metaheuristic algorithms in the forms of HICA, ICA, and SA for no-wait SDST-HFSP with ready time and machine availability to minimize the mean tardiness of jobs. The performance of the proposed algorithms was studied in terms of RPD. The results of the numerical experiments revealed that HICA outperformed the other algorithms.

3.6 SDST-MOHFSP (Metaheuristic of Neighborhood Searches)

The VNS has been first proposed by Mladenovic and Hansen [157]. It is relatively new metaheuristic framework for developing algorithms to solve combinatorial optimization problems [158]. VSN explores unacceptable neighborhoods of the current solution, and moves from there to a new improved solution. The LS method is applied repeatedly to get from solutions in the neighborhood to local optima. More precisely, VNS systematically changes the neighborhood in two phases: firstly, descent to find a local optimum and finally, a perturbation phase to get out of the corresponding valley. Concerning the use of metaheuristic VSN for solving scheduling problems, it can be divided into trajectory methods or LS heuristics and population-based methods. Population-based methods deal with a set of solutions in every iteration of the algorithm while trajectory methods only deal with a single solution. As one of the most cited population-based methods, GA shows robust performance with various scheduling problems. However, GA usually takes more computing efforts to locate the optimal in the region of convergence, owing to the lack of LS ability. Contrarily, the trajectory method, such as VNS, has shown its potential in manipulating the promising regions in the search space with superiority solutions. Nevertheless, it is still prone to premature convergence traps due to the limited manipulation ability [85]. With this regard, this section reviews the works dealing with application of the metaheuristic VNS for solving SDST-MOHFSP.

Naderi et al. [166] proposed a hybrid VNS combined with of VNS descent, to minimize the total completion time in a SDST-HFFSP. The same problem to minimize makespan is investigated in [167] and a VNS was proposed. All the results illustrated that the VNS outperformed the other algorithms. Shahvari et al. [207] developed six algorithms based on the TS for group scheduling in a SDST-HFFSP minimizing the makespan. The best of these algorithms was compared with the existing algorithm in the literature. Behnamian et al. [16] proposed three hybrid metaheuristics of SA and VNS for solving SDST-MOHFSP to minimize sum of the earliness and tardiness of jobs. Later Behnamian et al. [17] anticipated a SDST-MOHFSP with the objectives of minimizing the makespan and sum of the earliness and tardiness of jobs. A hybrid VNS algorithm was developed consisting of three phases. In another research Behnamian et al. [18] contemplated a machine scheduling problem in the form of SDST-MOHFSP with the objectives of minimizing the sum of earliness and tardiness. In this work three hybrid metaheuristic algorithms using VNS, ACO and a VNS were proposed.

Rashidi et al. [193] addressed a hybrid SDST-MOHFSP with unrelated parallel machines and processor blocking to minimize the makespan and the maximum tardiness. An algorithm consisting of independent parallel GA by dividing the whole population into multiple subpopulations was presented. Each subpopulation was assigned with different weights to search for optimal solutions in different directions. To further cover the Pareto solutions, each algorithm was combined with a LS step and a new procedure called ‘redirect’. For additional preventing the occurrence of the trap, a LS based on elite chromosomes was also incorporated in the algorithm steps. Majazi Dalfard et al. [145] presented a hybrid VNS-GA for an assembly SDST-MOHFSP with transportation times to minimize sum of the total weighted squared tardiness, makespan, total weighted squared earliness and the number of tardy jobs. Behnamian et al. [22] proposed a hybrid metaheuristic, comprising two components: GA; and VNS to solve a hybrid SDST-MOHFSP. A combination of a PSO and genetic operators for a SDST-MOHFSP that minimizes the mean weighted completion time and the sum of the weighted tardiness/earliness costs is proposed in [222]. The PSO was modified by introducing genetic operators, such as crossover and mutation operators, to update particles and improve particles by VSN.

An algorithm based on IG technique for solving a permutation SDST-MOHFSP with objectives involving the makespan, the tardiness and the flow time was presented in [155]. Mousavi et al. [161] addressed the scheduling problems in a hybrid SDST-MOHFSP with two objectives of minimizing the makespan and the total tardiness and designed four evolutionary algorithms based on the GA. Maleki Darounkolaei and Seyedi [147] adapted the similar problem and developed a VNS and an SA for the problem. To enhance the performance of the SA, its parameters were optimized by the use of Taguchi method. Mozdgir et al. [165] considered a two-stage SDST-MOHAFSP to minimize weighted sum of makespan and mean completion time. Lu and Logendran [137] investigated a SDST-MOHFSP assuming dynamic job releases and machine availabilities to minimize the weighted sum of total weighted completion time and total weighted tardiness. Two different initial solution finding mechanisms were proposed, and a TS-based two-level search algorithm was used to find solutions of the problem.

A Pareto-based iterated LS was developed in [232] by incorporating Pareto concept into the conventional ISL for the SDST-MOHFSP with objectives of the tardiness, the makespan and the flow time. In this algorithm, a variable depth search method based on dynamic neighborhood was designed. Eskandari and Hosseinzadeh [57] adapted a reentrant SDST-HFSP to minimize the makespan of the schedule. First, a MILP was presented for the problem. Then, several basic dispatching rules and a VNS were proposed. Hatami et al. [91] adapted a distributed SDST-HAFSP to minimize the makespan of jobs. Two heuristic algorithms and two metaheuristic algorithms were constructed by simplification of the IG and the VSN methodology. Campos et al. [37] studied a three-stage assembly SDST-HFSP to minimize the total tardiness of the orders. A heuristic based on general VNS combined with random LS procedure was constructed to obtain near-optimal solutions. A destruction-construction strategy based on IG was also developed for the solution diversity. Li and Li [130] focused on a SDST-MOHFSP with both sequence dependent setup times and sequence independent setup times to minimize makespan and total flow time. A multi-objective LS framework-based decomposition, called MOLSD was built by which a multi-objective problem was decomposed into a number of single objective optimization sub problems. To search the promising neighbors of each non-dominated solution, NEH algorithm was used. Then, a Pareto LS embedded with a heavy perturbation operator was applied to search the promising neighbors of each non-dominated solution. Finally, a single insert-based LS, a multiple LS strategy, and a doubling perturbation mechanism were designed to exploit the new individual.

Tian et al. [225] presented a Pareto-based adaptive bi-objective VSN (PABOVNS) to deal with the SDST-MOHFSP with the minimization of the total weighted tardiness and the total setup time. Two routines such as VNS, LS based on neighborhood search are combined as the solution procedure. Sukkerd and Wuttipornpun [216] proposed a dispatching heuristic and three non-population search algorithms for the capacitated material requirement planning in a multi-stage assembly SDST-HFSP with two alternative machines. The initial sequence was constructed by a simple due date-based dispatching rule and the sequence was repeatedly improved to reduce the total cost by applying a TS, an SA and a VNS separately. The result showed that VNS significantly outperformed TS, SA and the existing algorithm.

3.7 Summary of the Review

A comprehensive review and evaluation of many existing heuristics and metaheuristics for the multi-objective SDST-MOHFSP has been presented in this paper. The drawbacks and limitations in the current state of art have been explored and direction for the current research has been specified. The results of this survey are listed in the Tables 3-5. These tables aim to help related researchers to get insight to existing research efforts and deriving agenda for future research. Four essential issues are carried out in these tables. These are: 1) the method(s) being addressed, 2) the criteria defined for the problem, 3) the type of model contemplated and 4) the validation of the approaches. These issues are summarized in Table 3 through Table 5. Table 3 is devoted to the papers considering the SDST-HFSP. Table 4 is committed to the papers dealing with the SDST-HFFSP. And Table 5 is dedicated to the works considering the assembly SDST-HFSP. For every individual paper, authors of the references are sorted per publishing dates and are listed in the first column. The second column provides the type of solution approach(s). The types of criteria used in each paper are presented in the third column (the notations of this column are listed in the appendix of this paper, Table (1P)). Column four shows the category of model(s) focused for SDST-MOHFSP. Finally, the last column provides information regarding the validation of the solution methods.

Table 3. A Consolidated Review of HFSP.

Reference	Method(s)	Objective Functions	Flowshop Model	Comments
Cochran et al. [46]	MPGA	C_{max} and T_w	SDST-HFSP	Outperformed MOGA
Chang et al. [39]	Heuristic + GA	$S + E + L$	SDST in BOPP film SP	GA with different mutation methods were applied.
Lin & Liao [133]	CHA	T^{W}_{Max}	SDST-HFSP	compared with the optimal solutions
Chang et al. [40]	Heuristic	C_{max}	SDST-HFSP	Validation is performed
Lee and Asllani [126]	Some heuristics	C_{max}, S and N_T	HFSP	Validation is performed
Ponnambalam et al. [184]	TSP + MOGA	C_{max}, $\bar{F}$ and MIT	HFSP	Validation is performed
Brown et al. [33]	Exact	C_{max} *and* $\sum_j T_j$	HFSP	Validation is performed
Zandieh et al. [238]	GA + RKGA	C_{max}	SDST-HFSP	An IA is used to tackle complex problems
Tao and Rongqiu [219]	Heuristic	C_{max}	SDST-HFSP	Validation is performed
Ruiz and Maroto [196]	GA	C_{max} and Total ST	SDST-HFSP + machine eligibility	Several metaheuristics are used for performance evaluation.
Hendizadeh et al. [96]	GA	C_{max} and F	SDST-HFSP	Validation is performed
Tavakkoli-Moghaddam et al. [221]	Biological AIS	T_w and $\overline{C_w}$	HFSP	Outperformed MISA, MOIA, PS-NC, SPEAII and NSGAII
Song et al. [215]	MP & heuristic	C_{max}	SDST-HFSP	Validation is performed
Chang et al. [41]	Artificial chromosome imbeded in GA and NSGA-II	C_{max} and T_w	HFSP	Speedup GA convergence

Abiri et al. [1]	TS	C_{max}	SDST-HFSP	Results are compared with RKGA
Behnamian et al. [16]	SA + VNS, ACO + VNS, and SA + ACO + VNS	$E+T$	HFSP	Outperformed VNS
Behnamian et al. [17]	GA and LS combine with SA and a VNS	C_{max} and $(E+T)$	HFSP	Outperformed some existing algorithms
Behnamian et al. [18]	SA + VNS, ACO+ VNS, and SA + ACO + VNS	$E+T$	PHFSP, Due dates assignment	Outperformed VNS
Naderi et al. [170]	HAS + LSSA with Migration	C_{max} and T_{max}	PHFSP	Outperformed some existing algorithms
Naderi, et al. [169]	HSA	C_{max} and T	HFSP +TT	Parameters are set by Taguchi
Qian et al. [185]	Hybrid DE + LS	C_{max}, L, $E+T$	SDST-MOHFSP	Pareto dominance is used
Naderi et al. [169]	HSA	C_{max} and T	SDST-MOHFSP with TT	Parameters are set by Taguchi
Mansouri et al. [148]	MOGA + SA + MOSA	C_{max} and S	SDST-MOHFSP	Outperformed MOSA
Zandieh and Gholami [240]	ILS	C_{max}	SDST-HFSP and random breakdown	In computational method immune response is used.
Zandieh and Rashidi [239]	HGA	C_{max}	SDST-HFSP and blocking	performance is compared with RKGA.
Gholami et al. [77]	GA + SM	C_{max}	SDST-HFSP and random breakdown	Validation is performed
Yaurima et al. [235]	GA	$\sum_j C_j$	SDST-HFSP and limited buffer	Validation is performed

Abiri et al. [1]	TS + GA	C_{max}	SDST-HFSP	The results are compared with RKGA
Alfieri [9]	SM + TS	T_{max} T_w and S	Reentrant SDST-MOHFSP	Applied to a real-world problem
Tabrizi et al. [217]	SA	C_{max}	SDST-HFSP	The superiority of SA is inferred
Luo et al. [141]	GA	C_{max}	SDST-HFSP with blocking + machine availability	Two types of machine unavailability is considered
Ramezanian et al. [191]	Exact + GA	$E + T\ cost$	SDST-MOHFSP	Validation is performed
Rashidi et al. [193]	Parallel GA combined with a local search	C_{max} and T_{max}	HUMFP	Validation is performed
Behnamian et al. [19]	Hybrid VNS + ACO + SA	$E + T$	SDST-HFSP	Compared with RKGA and IA
Behnamian et al. [20]	HGA +LS	C_{max} & $(E + T)$	SDST-MOHFSP	Validation is performed
Li et al. [129]	Exact + HGA +LS	C_{max} + E	SDST-MOHFSP	Validation is performed
Xiao and Zheng [231]	Exact	C_{max}	SDST-HFSP	Model is implemented in a semiconductor ATM
Eren [56]	Modified NEH	C and C_{max}	SDST-MOHFSP	Validation is performed
Behnamian et al. [22]	GA + VNS	Min–max and weighted techniques	SDST-MOHFSP	Validation is performed
Behnamian and Fatemi-Ghomi [23]	Hybrid VNS + ACO + SA	C_{max} and TRC	Proportionate HFSP	Outperformed a variable neighborhood search and a GA

Behnamian and Zandieh [21]	Exact	E and T^2	SDST-MOHFSP	Validation is performed
Hekmatfar et al. [95]	13 heuristics including HGA and RKGA	C_{max}	SDST-HFSP	HGA outperforms the others
Tavakkoli-Moghaddam et al. [222]	HPSO + GA	$\sum_j (w_j T_W + v_j E_W)$ and $\bar{C}$	SDST-MOHFSP	Outperformed MOGA
Mousavi et al. [160]	BOH	C_{max} and T	SDST-MOHFSP	Compared with SA & VNS
Mirsanei et al. [156]	SA	C_{max}	SDST-HFSP	Compared with RKGA and IA.
Minella et al. [155]	IG	C_{max}, F and T	PHFSP	Outperformed some existing algorithms
Behnamian et al. [24]	ACO + SA + VNS	C_{max}	SDST-HFSP	Compared the results with the several cited algorithms
Gómez-Gasquet et al. [79]	Agent-based GA	C_{max}	SDST-HFSP	Validation is performed
Javadian et al. [100]	ILS	C_{max}	SDST-HFSP with time lag	Compared with MIP and a cited algorithm.
Gupta et al. [84]	Heuristic	C_{max}	SDST-MOHFSP	Validation is performed
Khalili and Tavakkoli-Moghaddam [115]	EA +SA	C_{max} and T_w	SDST-MOHFSP	Validation is performed
Hakimzadeh Abyaneh and Zandieh [87]	GA	C_{max} and T	SDST-MOHFSP limited buffers	Validation is performed
Lin and Ying [134]	A two-level multi-start SA	C_{max} and T	SDST-MOHFSP	Outperformed some existing algorithms

Mousavi et al. [161]	Several GA based	C_{max} and T	SDST-MOHFSP	Outperformed PGA,-ALS and MOGLS
Gicquel et al. [75]	Exact, B&C	T_W	SDST-MOHFSP	Validation is performed
Tavakkoli-Moghaddam and Amin-Tahmasbi [223]	MOIS	and C_{max} $\alpha_1 E + \alpha_2 T$	SDST-MOHFSP	Validation is performed
Seidgar et al. [203]	GA + ICA	$\alpha_1 C_{max} + \alpha_2 T$	SDST-HFS	Validation is performed
Wang et al. [229]	Exact + LS	C_{max} and N_T	SDST-MOHFSP	Validation is performed
Wang et al. [230]	HPSO	C_{max} and N_T	SDST-MOHFSP HFS with lot streaming	Compared with MILP.
Rezaeian et al. [194]	GA + ICA	C_{max} and T_{max}	SDST and preemption	Validation is performed
Fadaei and Zandieh [58]	MPGA	C_{max} and T	HFP	Validation is performed
Behnamian and Zandieh [25]	Hybrid PSO + SA + VNS	$E + T$	SDST-MOHFSP	Validation is performed
Lu and Logendran [137]	TS	$\sum_j (w_j T_W + v_j C_W)$	SDST-MOHFSP	Validation is performed
Hecker et al. [93]	Several EA	C_{max}	SDST-MOHFSP	Validation is performed
Ciavotta et al. [45]	Restarted IPG	C_{max} and T	PFP	Outperformed some existing algorithms
Xu et al. [232]	IPG + LS	C_{max} and T	PFP	Outperformed some existing algorithms
Behnamian et al. [26]	*Lp*-metric + PSO +VNS	C_{max} and $(E + T)$	SDST-MOHFSP	Validation is performed
Behnamian [27]	CCA + VNS	C_{max}, $(E + T)$ and cost	SDST-MOHFSP	Validation is performed

Hecker et al. [94]	Modified GA + A	C_{max}	SDST-MOHFSP	Validation is performed
Eskandari and Hosseinzadeh [57]	ILS	C_{max}	re-entrant SDST-HFSP	Validation is performed
Ebrahimi et al. [55]	NSGAII + MOGA.	C_{max} and T	SDST-MOHFSP	Validation is performed
Saravanan et al. [200]	AIS	C_{max}, E, C and T	SDST-MOHFSP	Outperformed SA and B-Grasp approach
Kayvanfar et al. [114]	Exact	C_{max}, E, and T	SDST-MOHFSP	Validation is performed
Sheikh et al. [210]	MILP	C_{max}, MSL	SDST-MOHFSP	Validation is performed
Lee et al. [127]	Hybrid BS + NEH	TWU	SDST-MOHFSP	Compared with existing LS method
Luo et al. [142]	GA	S and C_{max}	SDST-MOHFSP	Validation is performed
Amirian and Sahraeian [10]	AEC	C_{max}, T_{max} and F	SDST-MOHFSP	Outperformed some existing algorithms
Gholami and Rajaee-Abyaneh [78]	Hybrid MA + PSO combined with GA	$E + T$	SDST-MOHFSP	Outperformed HAS
Rajaee Abyaneh and Gholami [189]	PSO + GA	$E + T$	SDST-MOHFSP	Parameters are set by Taguchi
Sioud et al. [213]	Hybrid GA + AIS	C_{max} and T	SDST-MOHFSP	Outperformed NSGA-II
Allahverdi [8]	Literature review			
Mohammadi [159]	GA	C_{Max}, N_T, E_W T_W	SDST-MOHFSP	Compared with LPT, SPT, EDD and NEH
Dhingra [53]	HSA	$C_{Max}, T_W^2, E_W^2, N_T$	SDST-MOHFSP	Outperformed some existing algorithms

Liou and Hsieh [135]	Hybridized PSO + GA	C_{max}	SDST-HFSP with TT	Compared with PSO and GA.
Li and Li [130]	MOLS	C_{Max}, and F	SDST-MOHFSP	Outperformed hyper-volume indicator; and set coverage
Satyanarayana and Pramiladevi [201]	Three HGA	$\sum_j (w_j T_W^2 + v_j C_{Max} + t_j E_W^2 + u_j N_T)$	SDST-MOHFSP	Outperformed [224]
Lee and Bang [128]	B&B and two-stage heuristic algorithm	T	SDST-HFSP	Validation is performed
Lu et al. [138]	MODGWO	ML and PE	SDST-MOHFSP	Outperformed NSGA-II and SPEA 2
Saravanan et al. [200]	AIS	C_{max}, E, C and T	SDST-MOHFSP	Outperformed SA and B-Grasp approach
Mousavi and Zandieh [162]	Hybrid SA + GA + LS	C_{max} and T	SDST-MOHFSP	Validation is performed
Harbaoui et al. [88]	GA	C_{max}	SDST-HFSP with time lags	Validation is performed
Mousavi et al. [163]	GA	C_{Max}, and T	Re-entrant SDST-MOHFSP	Outperformed Meshless Local Petrov Galerkin Algorithm
Tian et al. [225]	Pareto-based AVNS	T_W and S	SDST-MOHFSP	Outperformed existing algorithms
Li and Ma [131]	MOSA	C_{Max}, and F	SDST-MOHFSP	Outperformed some existing algorithms
Lu et al. [139]	HMOBSA	C_{Max} and ML	SDST-MOHFSP	Outperformed other methods
Lu et al. [140]	MODGWO	C_{Max}, ML and MI	SDST-MOHFSP	Outperforms NSGA-II, SPEA2

Pan et al. [176]	LS and Greedy heuristics, FFO, MBO, ABCO	C_{max}	SDST-HFSP	Validation is performed
Yazdania and Naderi [236]	VNS and GA	C_{max}	SDST-HFSP	Validation is performed
Li & Ma [132]	MODABCA	C_{Max}, and F	SDST-MOHFSP	Compared with some cited algorithms
Sharma et al. [208]	DR	C_{Max}, OPC	SDST-MOHFSP	Validation is performed
Sharma et al. [209]	DR	C_{Max} and MRC	SDST-MOHFSP with job blocking	Validation is performed
Pramiladevi [202]	Five GAs	C_{max} and T^2	SDST-MOHFSP	Compared with [224]
Xu et al. [234]	ILS	C_{Max} and T_{Max}	SDST-MOHFSP	Outperformed MOEAs
Mohammadi [159]	HICA + ICA +SA	$\bar{T}$	SDST-HFSP	outperforms ICA and SA
Satyanarayana and Pramiladevi [202]	Five HGA	C_{Max} and $\sum_j (w_j T_W^2)$	SDST-MOHFSP	Outperformed [224]
Mousavi et al. [164]	VNS based priority	C_{Max}, and T	SDST-MOHFSP	Validation is performed

Table 4. A Consolidated Review of Flexible HFSP.

Reference	Method(s)	Objective Functions	Flowshop Model	Comments
Kurz [123]	Heuristics and RKGA	C_{max}	SDST-HFFSP	Validation is performed
Liu and Chang [136]	LR-based approach + ILS	S and S_C	SDST-HFFSP	Validation is performed
Cheng et al. [41]	GH, IHc	C_{max}	SDST-HFFSP	Validation is performed
Kurz [123]	LR, ILS	S and S_C	SDST-MOHFFSP	Validation is performed
Kurz and Askin [124]	GH + IH +adaptation of Johnson's rule	C_{max}	SDST-HFFSP	Validation is performed
Kurz and Askin [125]	GH + IH +RKGA	C_{max}	SDST-HFFSP	A lower bound has been created
Ponnambalam et al. [184]	Evolutionary search for TSP + GA	C_{Max}, $\overline{F}$ and MIT	SDST-MOHFFSP	Validation is performed
Jungwattanakit et al. [107]	IH +Two CHA + TS + SA + GA	$\lambda_1 w_1 C_{Max} + \lambda_2 w_2 N_T$ where $0 \le \lambda_1, \lambda_2 \le 1$ $\lambda_1 + \lambda_2 = 1$	SDST-MOHFFSP	Outperforms the other proposed algorithms
Jungwattanakit et al. [107]	CHA + TS + SA + GA	$\lambda_1 w_1 C_{Max} + \lambda_2 w_2 N_T$ where $0 \le \lambda_1, \lambda_2 \le 1$ $\lambda_1 + \lambda_2 = 1$	SDST-MOHFFSP	Outperforms SA SA outperforms TS TS outperforms GA
Crowder [48]	CHA + SA	C_{max}	SDST-HFFSP limited buffer	A lower bound is generated

Tavakkoli-Moghaddam, and Safaei [220]	Exact, Classical model and heuristic	C_{max}	SDST-HFFSP and blocking processors	Validation is performed
Naderi et al. [166]	VNS	$\sum_j C_j$	SDST-HFFSP	Compared against with some cited algorithms
Allahverdi [6]	Literature review			
Ruiz et al. [197]	IGH	C_{max} and $\sum_j w_j T_j$	SDST-MOHFFSP	Validation is performed
Jungwattanakit et al. [108]	GA + CHA	$\lambda C_{Max} + (1-\lambda) N_T$ where $0 \le \lambda \le 1$	SDST-MOHFFSP	Compared with (Allahverdi and Soroush 2008)
Joli et al. [103]	MA	C_{max}	SDST-HFSP	Compared with several cited algorithms
Maboudian and Shafaei [144]	Exact	C_{max} and T_{max}	two-stage SDST-MOHFFSP	Validation is performed
Zandieh et al. [241]	SA and GA	C_{max}	SDST-HFFSP	Validation is performed
Naderi et al. [168]	IG	$\sum_j w_j C_j$	SDST-HFFSP	Validation is performed
Jungwattanakit et al. [109]	SA + TS + GA	$\lambda C_{Max} + (1-\lambda) N_T$ where $0 \le \lambda \le 1$	SDST-MOHFFSP	Outperforms SA, TS and GA
Naderi et al. [167]	VNS	C_{max}	SDST-HFFSP	Validation is performed
Shahvari et al. [207]	TS	C_{max}	SDST-HFFSP	Validation is performed
Ashrafi et al. [12]	GRASP	$\sum_j C_j$ and T_{max}	SDST-MOHFFSP	Validation is performed
Tang and Song [218]	discrete PSO	C_{max}	SDST-HFFSP	Validation is performed

Urlings et al. [227]	GA	C_{max}	SDST-HFFSP	Validation is performed
Kia et al. [117]	GA + seven DR + CHA	$\bar{F}$ and $\bar{T}$	SDST-MOHFFSP	Compared with six existing algorithms
Naderi et al. [171]	dynamic DR + ILS	C_{max}	SDST-HFFSP	Compared with seven existing algorithms
Zandieh et al. [242]	GA	C_{max} and computational time	SDST-MOHFFSP with time lag	Compared with SPT, LPT & NEH.
Karmakar and Mahanty [113]	Exact, GA and theory of constraint	C_{max}	SDST-HFFSP and infinite intermediate storages	Compared with six existing algorithms
Urlings et al. [227]	GA	C_{max}	SDST-HFFSP with machine release,, time lags	Validation is performed
Karimi et al. [111]	MPGA	$\sum_j (w_j T_W + v_j C_{Max})$	SDST-MOHFFSP	Parameters are set by DOE
Zandieh and Karimi [243]	MPGA	$\sum_j (w_j T_W + v_j C_{Max})$	SDST-MOHFFSP	Outperformed NSGA-II
Javadian et al. [100]	Exact + IA	C_{max}	SDST-HFFSP	Validation is performed
Kianfar et al. [119]	HGA + SM	$\bar{T}$	Stochastic SDST-HFFSP	Validation is performed
Jolai et al. [104]	PBSA + AICA and combined AICA- PBSA	C_{max}	SDST-HFFSP	Parameters are set by Taguchi
Acosta et al. [3]	GA	C_{max} and Due date	SDST-MOHFFSP	Validation is performed
Rezaeian et al. [194]	GA + ICA + HGA + HICA	$\alpha_1 C_{max} + \alpha_2 T_{max}$	SDST-MOHFFSP	Validation is performed
Sioud et al. [211]	LS	$\sum_j T_j$	SDST-HFFSP	Validation is performed
Sioud et al. [212]	GA + ACO + HGA	C_{max}	SDST-HFFSP	Validation is performed

Saluja and Jain [198]	GA	C_{max}	SDST-HFS lot splitting	Validation is performed
Attar et al. [13]	PSO +SPEA	C_{Max}, and T	SDST-MOHFFSP	Validation is performed
Tyagi et al. [226]	Exact	$\alpha_1 C_{max} + \alpha_2 T_{max}$	SDST-MOHFFSP	Validation is performed
Rabiee et al. [186]	ICA + GA +ACO + VNS	C_{max}	Reentrant SDST-HFFSP	Parameters are set by Taguchi
Panahi [177]	Exact	C_{max} and T	SDST-MOHFFSP	Parameters are set by Taguchi
Zandieh and Hashemi [244]	GA	C_{max}	SDST-HFFSP	Parameters are set by Taguchi
Ramezani et al. [190]	IWO +VSN + SA	C_{max}	SDST-HFFSP	Parameters are set by Taguchi
Kangarloo et al. [110]	GA + ICA	$\alpha_1 \sum_j w_j T_j + \alpha_2 \sum_j w_j E_j$	SDST-MOHFFSP with machine breakdown	Validation is performed
Hosseini [98]	Exact	C_{max}	SDST-HFFSP	Validation is performed
Lu et al. [138]	NSGA-II + SPEA2	*PE* and *ML*	SDST-MOHFFSP with job dependent	Validation is performed
Kia et al. [118]	Four DR	$\overline{F}$ and $\overline{T}$	SDST-MOHFFSP	Validation is performed
Sivapragasam and Suppiah [214]	GA	$\sum_j w_j T_j$	SDST-HFFSP	Compared with some GAs
Ramezaninian et al. [192]	Exact + Heuristics	C_{max}	Capacity SDST-HFFS	Validation is performed

Table 5. A Consolidated Review of Assembly HFSP

Reference	Method(s)	Objective Functions	Flowshop Model	Comments
Yokoyama [237]	B&B	$\sum_j w_j C_j$	SDST-HAFSP	Validation is performed
JIN et al. [101]	GA	C_{max}	SDST-HAFSP	Validation is performed
Rahimi-Vahed and Mirzaei [187]	FLA + BO	S, TWU, and TTV	SDST-MOHAFSP	Outperformed PS-NC GA, NSGA-II, and SPEA-II
Hatami et al. [90]	SA + TS	$\bar{F}$ and T_{max}	Three-stage SDST-MOHAFSP	Validation is performed
Majazi Dalfard et al. [145]	HGA +VNS	$\alpha_1 C_{max} + \alpha_2 \sum_j w_j {T^2}_j + \alpha_3 \sum_j w_j {E^2}_j + \alpha_4 N_T$	SDST-MOHAFSP	Validation is performed
Fadaei and Zandieh [58]	SA	C_{max}	SDST-HAFSP	Validation is performed
Pargar and Zandieh [180]	WFA	$\sum_j (w_j T + v_j C_{Max})$	SDST-MOHAFSP	Outperformed RKGA
Seyedi et al. [205]	SA + TS	$\alpha_1 C_{max} + \alpha_2 \bar{C}$	SDST-MOHAFSP with block time	Validation is performed
Maleki-Darounkolaei et al. [146]	SA	C_{max} and $\sum_j w_j C_j$	SDST-MOHAFSP	Validation is performed
Mozdgir et al. [165]	Exact + VNS	$\alpha_1 C_{max} + \alpha_2 \bar{C}$	Two-stage SDST-MOHAFSP	Validation is performed

Fattahi et al. [60]	Exact + Johnson's algorithm	C_{max}	SDST-HAFSP	Two lower bounds are introduced
Fattahi et al. [61]	GA + SA + NEH and Johnson's algorithm	C_{max}	SDST-HAFSP	Three lower bounds are presented
Maleki-Daronkolaei and Seyedi [147]	VNS + SA	$\alpha_1 C_{max} + \alpha_2 \bar{C}$	SDST-MOHAFSP	Validation is performed
Campos and Arroyo [38]	NSGA-II coupled with the IG	$\sum_j T_j$ and F	Three-stage SDST-MOHAFSP	Outperformed GRASP
Fattahi et al. [62]	Hierarchical B&B	C_{max}	SDST-HAFSP	Some lower and upper bounds are developed
Allahverdi, and Aydilek [7]	GA +, two SA + IA	$\sum_j T_j$	SDST-HAFSP	Validation is performed
Komaki et al. [120]	CS	C_{max}	SDST-HAFSP	Validation is performed
Khalili and Naderi [116]	ICA	$C_{max}\ and\ \sum_j T_j$	No-wait SDST-MOHAFSP	Validation is performed
Komaki et al. [120]	CS + VNS	C_{max}	Multi-layer SDST-HAFSP	Validation is performed
Hatami et al. [91]	IG + VNS	C_{max}	SDST-HAFSP	Validation is performed
Nejati et al. [175]	GA + SA	$\sum_j w_j C_j$	SDST-HAFSP	Validation is performed
Hosseini [98]	Exact + Johnson's algorithm	C_{max}	SDST-HAFSP	Validation is performed
Du et al. [54]	PSO + SA	C_{max}	SDST-HAFSP	Validation is performed
Campos et al. [37]	VSN combined with random LS	$\sum_j T_j$	Three-stage SDST-HAFSP	Validation is performed

Sukkerd, and Wuttipornpun [216]	TS + SA + VNS	$\sum(F,E,T)$	SDST-HAFSP	Validation is performed
Fattahi et al. [62]	Hierarchical B&B	C_{max}	SDST-HAFSP	Some lower and upper bounds are developed
Allahverdi, and Aydilek [7]	GA +, two SA + IA	$\sum_j T_j$	SDST-HAFSP	Validation is performed
Komaki et al. [120]	CS	C_{max}	SDST-HAFSP	Validation is performed
Khalili and Naderi [116]	ICA	C_{max} *and* $\sum_j T_j$	No-wait SDST-MOHAFSP	Validation is performed
Komaki et al. [120]	CS + VNS	C_{max}	Multi-layer SDST-HAFSP	Validation is performed
Hatami et al. [91]	IG + VNS	C_{max}	SDST-HAFSP	Validation is performed
Nejati et al. [175]	GA + SA	$\sum_j w_j C_j$	SDST-HAFSP	Validation is performed
Hosseini [98]	Exact + Johnson's algorithm	C_{max}	SDST-HAFSP	Validation is performed
Du et al. [54]	PSO + SA	C_{max}	SDST-HAFSP	Validation is performed
Campos et al. [37]	VSN combined with random LS	$\sum_j T_j$	Three-stage SDST-HAFSP	Validation is performed
Sukkerd, and Wuttipornpun [216]	TS + SA + VNS	$\sum(F,E,T)$	SDST-HAFSP	Validation is performed

4. Concluding Remarks and Future Research Recommendations

In this survey, a nomenclature has been formed which is used to classify existing research works. Furthermore, this review aims to help the researcher in main ideas, and theory, the methodological approach, and the gaps to fill of the SDST-MOHFSP. The following subsections conclude this survey and present vast list recommendations for future research.

5. Conclusion and Feature Reasearch

This paper, presents a comprehensive study for reviewing the most significant approaches for solving SDST-MOHFSP. A preliminary investigation discloses that exact solution approaches are still incapable of solving real-world applications. Due to this fact our review is focused on use of heuristic and metaheuristic approaches for solving SDST-MOHFSPs. In general, with this survey we aim to point out current trends, gaps, and directions for future research and to facilitate the searching process on papers dealing with the SDST-MOHFSP. To attain these goals a detailed presentation of the theoretical background of FSP and the setup time concepts, as well as a category of multi-objective FSP along with a classification of the FSP are studied. Then the existing research works are classified per the methods while for each category the problem spec., the criteria applied, the applied methodology, and the evaluation of the approach validation are addressed.

According to the results of this survey it is evident that the research area on SDST-MOHFSP has received a considerable devotion over the last decades and significant advancement has been accomplished. The results also reveal a number of observations which help to direct future research efforts. Differential evolution (DE) is commonly known as metaheuristic which makes few or no assumptions about the problem being optimized and can search over the very large spaces of candidate solutions. Also, artificial intelligent (AI) tools like artificial neural network are found to be extremely useful in modeling reliable processes in the field of selecting optimal parameters during scheduling optimization. Considering the solution approaches cited for SDST-MOHFSP, it is shown that works dealing with the use of DE and AI is missing. Therefore, applying these methods for the under reviewed problems are recommended. A thorough review of the literature indicates that although in the majority of the research works a mathematical programming model has been presented, but no work has considered the type of the decision variables that we have proposed. Therefore, as a further research, it would be beneficial to evaluate the computational efficiency of the proposed mathematical programming model for solving SDST-MOHFSPs. The multi-objective FSP can be categorized per different assumptions such as assumptions concerning machines; assumptions concerning job; assumptions concerning operating policies, etc. Based on the results of the literature review, a considerable verity of the SDST-MOHFSP models, which are very useful to the real -world problems have not been addressed. Therefore, they could be interesting subjects for the further studies. These are:

- The SDST-MOHFSP model with preemptive resume operations where the setup time is relatively small comparing to the operation processing time.

- The SDST distributed flowshop in which for assigning operations there is more than one alternative factory and the setup times are differed on the selected factory machine.
- The SDST-MOHFSP model with allowing unavailability of machines due to breakdown or preventive maintenance.
- The SDST-MOHFSP model with the anticipatory sequence dependent setup times.

Beside manufacturing environment there are many situations in which the scheduling problems can be modeled in the form of a FSP. These include; consulting; research and development; industrial design companies; and so on. These types of firms are generally dealing with a set of projects, each with the multiple deliverable phases, in which phases have usually unidirectional precedence structure and there is a penalty associated for each tardy delivered phase. Considering each project as a job, the set of tasks to be performed by each department as an operation, and each department as a machine, we may have a FSP with the intermediate due dates. Upon assigning each operation to the specialist in each department, there are some prerequisite operations to become familiar with the spec of that particular project. Considering the operation processing times of the prerequisite as the setup times, the setup times may vary according to the previously assigned project. Therefore, we have a sequence dependent setup FSP. Furthermore, considering one or some completion time related criteria as well as one or some due dates related criteria simultaneously, we encounter with a wide variety of SDST-MOHFSPs. A thorough investigation of the related literatures reveals that there is no single work directing on the study of SDST-MOHFSPs with the intermediate due dates. Therefore, a large number of further researches are open to be explored for these types of problems. These are:

- Exploring different heuristic or metaheuristic methods for the SDST-MOHFSP with the intermediate due dates.
- Exploring different heuristic or metaheuristic methods for the SDST-MOHFSP with the intermediate due dates, and additional assumptions concerning machines; and operating policies.
- Exploring different criteria for optimizing the SDST-MOHFSP with the intermediate due dates.
- Exploring different criteria for optimizing the SDST-MOHFSP with the intermediate due dates, and additional assumptions concerning machines; and operating policies.

References

[1] Abiri, M., Zandieh, M., & Tabriz, A. A. (2009). A tabu search approach to hybrid flow shops scheduling with sequence-dependent setup times. *Journal of applied sciences*, *9*(9), 1740-1745.
[2] Abyaneh, S. H., & Zandieh, M. (2012). Bi-objective hybrid flow shop scheduling with sequence-dependent setup times and limited buffers. *The Int. J. of Ad. Manuf. Tech*, *58*(1-4), 309-325.
[3] Acosta, J. H. T., González, V. A. P., & Bello C. A. L. (2013). A genetic algorithm for simultaneous scheduling problem in flexible flow shop environments with unrelated parallel machines, setup time and multiple criteria. *Int. Conference on Advanced Manuf. Engineering and Tech*.
[4] Agnetis, A., Pacifici, A., Ross, F., Lucertini, M., Nicoletti, S., Nicolo, F., & Pesaro, E. (1997). Scheduling of flexible flow lines in an automobile assembly plant. *Eur. J. of Operational Res.* *97*(2), 348-362.

[5] Alaei, R., & Ghassemi-Tari, F. (2011). Development of a genetic algorithm for advertising time allocation problems. *Journal of industrial & systems engineering*, *4*(4), 245-255.
[6] Allahverdi, A., Ng, C., Cheng, T.E., Kovalyov, M.Y. (2008). A survey of scheduling problems with setup times or costs. *European journal of operational research, 187*(3), 985-1032.
[7] Allahverdi, A., & Aydilek, H. (2015). The two stage assembly flowshop scheduling problem to minimize total tardiness. *Journal of intelligent manufacturing, 26*(2), 225-237.
[8] Allahverdi, A, (2015). The third comprehensive survey on scheduling problems with setup times/costs. *European journal of operational research, 246*(2), 345-378.
[9] Alfieri, A, (2009). Workload simulation and optimisation in multi-criteria hybrid flowshop scheduling: a case study. *International journal of production research, 47*(18), 5129-5145.
[10] Amirian, H., Sahraeian, R. (2015). Augmented ε-constraint method in multi-objective flowshop problem with past sequence set-up times and a modified learning effect. *Int. J. of Production Res. 53*(19), 5962-5976.
[11] Andrés, C., Albarracín, J. M., Tormo, G., Vicens, E., & García-Sabater, J. P. (2005.) Group technology in a hybrid flowshop environment: A case study. *European J. of Operational Res, 167*(1), 272-281.
[12] Ashrafi, M., Davoudpour, H., & Abbassi, M. (2014). Investigating the efficiency of GRASP for the SDST HFS with controllable processing times and assignable due dates. *Handbook of research on novel soft computing intel. Algorithms: Theory and Pract. Appl* (pp. 538-567). IGI Global, 538-567.
[13] Attar, S., Mohammadi, M., Tavakkoli-Moghaddam, R., Yaghoubi, S. (2014). Solving a new multi-objective hybrid flexible flowshop problem with limited waiting times and machine-sequence-dependent set-up time constraints. *Int. J. of Comp. Integrated Manuf, 27*(5), 450-469.
[14] Azab, SS., & Hefny, HA. (2017). Swarm intelligence in semi-supervised classification. *Data mining and knowledge engineering. 9*(5): 99-103.
[15] Barabási, B. A. L., & Bonabeau, E. (2003). Scale-free. *Scientific american, 288*(5), 50-59.
[16] Behnamian, J., Zandieh, M., Fatemi Ghomi, S. M. T. (2009a). Parallel-machine scheduling problems with sequence-dependent setup times using an ACO, SA and VNS hybrid algorithm. *Expert systems with applications, 36*(6), 9637-9644.
[17] Behnamian, J., Fatemi Ghomi, S. M. T., & Zandieh, M. (2009b). A multi-phase covering Pareto-optimal front method to multi-objective scheduling in a realistic hybrid flowshop using a hybrid metaheuristic. *Expert systems with applications, 36*(8), 11057-11069.
[18] Behnamian, J, Zandieh, M., & Fatemi Ghomi, S. M. T. (2009c). Due window scheduling with sequence-dependent setup on parallel machines using three hybrid metaheuristic algorithms. *The Int. J. of Ad. Man.Tech, 44*(7-8):795-808.
[19] Behnamian, J., Fatemi Ghomi, S. M. T., & Zandieh, M. (2010a). Development of a hybrid metaheuristic to minimise earliness and tardiness in a hybrid flowshop with sequence-dependent setup times. *International journal of production research, 48*(5), 1415-1438.
[20] Behnamian, J., Zandieh, M., & Fatemi Ghomi, S. M. T. (2010b). A multi-phase covering Pareto-optimal ront method to multi-objective parallel machine scheduling. *Int. J. of Prod. Res, 48*(17), 4949-4976.
[21] Behnamian, J., & Zandieh, M. (2011). A discrete colonial competitive algorithm for hybrid flowshop scheduling to minimize earliness and quadratic tardiness penalties. *Expert Sys. with Appl, 38*(12), 14490-14498.
[22] Behnamian, J., Zandieh, M., & Fatemi Ghomi, S. M. T. (2011). Bi-objective parallel machines scheduling with sequence-dependent setup times using hybrid metaheuristics and weighted min–max technique. *Soft computing, 15*(7), 1313-1331.
[23] Behnamian, J., & Fatemi Ghomi, S.M.T. (2011). Hybrid flowshop scheduling with machine and resource-dependent processing times. *Applied mathematical modelling, 35*(3), 1107-1123.
[24] Behnamian, J., Fatemi Ghomi, S. M. T., & Zandieh, M. (2012). Hybrid flowshop scheduling with sequence-dependent setup times by hybridizing max–min ant system, simulated annealing and variable neighbourhood search. *Expert systems, 29*(2), 156-169.
[25] Behnamian, J., & Zandieh, M. (2013). Earliness and tardiness minimizing on a realistic hybrid flowshop scheduling with learning effect by advanced metaheuristic. *Arabian J. for Sc. & Eng*. 1-14.

[26] Behnamian, J., Fatemi Ghomi, S. M. T., & Zandieh, M. (2014). Realistic variant of just-in-time flowshop scheduling: Integration of Lp-metric method in PSO-like algorithm. *The Int. J. of Ad. Man.Tech, 75*(9-12), 1787-1797.

[27] Behnamian, J. (2014). Scheduling and worker assignment problems on hybrid flowshop with cost-related objective function. *The Int. J. of Ad. Manuf. Tech, 74*(1-4), 267-283.

[28] Blum, C., Roli, A., & Sampels, M. (2008). *Hybrid metaheuristics: an emerging approach to optimization (Vol. 114)*. Springer.

[29] Blum, C., Puchinger, J., Raidl, G. R., & Roli, A. (2011). Hybrid metaheuristics in combinatorial optimization: A survey. *Applied soft computing*, *11*(6), 4135-4151.

[30] Bonabeau, E., Dorigo, M., Theraulaz, G. (1999). *Swarm intelligence: from natural to artificial systems*. Oxford university press.

[31] Bozorgirad, M. A., & Logendran, R. (2016). A comparison of local search algorithms with population-based algorithms in hybrid flow shop scheduling problems with realistic characteristics. *The international journal of advanced manufacturing technology, 83*(5-8): 1135-1151.

[32] Brintha, N., Benedict, S., & Jappes, J. W. (2017). A Bio-Inspired Hybrid Computation for Managing and Scheduling Virtual Resources using Cloud Concepts. *Appl. math, 11*(2),565-572.

[33] Brown, S., McGarvey, R., & Ventura, J. (2004). Total flowtime and makespan for a no-wait m-machine flowshop with set-up times separated. *J. of the Operational Res. Society, 55*(6), 614-621.

[34] Burtseva, L, Yaurima, V, Parra, RR. (2010). Scheduling methods for hybrid flow shops with setup times. T. Aized, Ed, *Future Manufacturing Systems* (pp. 137-162). Croatia: Sciyo.

[35] Calvet, L., Armas, J. D., Masip, D., & Juan A. A. (2017). Learnheuristics: hybridizing metaheuristics with machine learning for optimization with dynamic inputs. *Open mathematics, 15*(1), 261-280.

[36] Campbell, H. G., Dudek, R. A., & Smith, M. L. (1970). A heuristic algorithm for the n job, m machine sequencing problem. *Management science, 16*(10), B-630-B-637.

[37] Campos, S. C., Arroyo, J. E. C., & Tavares, R. G. (2017). A general VNS heuristic for a three-stage assembly flow shop scheduling problem. In Madureira, A., Abraham, A., Gamboa, D., & Novais, P. (Eds). *Intel.Sys. Design and Appl*. *Advances in Intelligent Systems and Computing*. Springer, Cham.

[38] Campos, S. C., & Arroyo, J. E. C. (2014). NSGA-II with iterated greedy for a bi-objective three-stage assembly flowshop scheduling problem. *Proceedings of the 2014 annual conference on genetic and evolutionary computation*ly. Vancouver, Canada.

[39] Chang, P. C., Hsieh, J. C., Wang, Y. W. (2003). Genetic algorithms applied in BOPP film scheduling problems: minimizing total absolute deviation and setup times. *App. Soft Comp, 3*(2), 139-148.

[40] Chang, J., Yan, W., & Shao, H. (2004). Scheduling a two-stage no-wait hybrid flowshop with separated setup and removal times. *American control conference*. Boston, MA, USA.

[41] Chang, P. C., Chen, S. H., Fan, C. Y., & Chan, C. L. (2008). Genetic algorithm integrated with artificial chromosomes for multi-objective flowshop scheduling problems. *Appl. Math. and Comp, 205*(2), 550-561.

[42] Chiandussi, G., Codegone, M., Ferrero, S., & Varesio, F. E. (2012). Comparison of multi-objective optimization methodologies for engineering applications. *Computers & mathematics with applications, 63*(5), 912-942.

[43] Cho, H. M., & Jeong, I. J. (2017). A two-level method of production planning and scheduling for bi-objective reentrant hybrid flow shops. *Computers & industrial engineering*, *106*, 174-181.

[44] Choi, B. C., & Park, M. J. (2016). An ordered flow shop with two agents. *Asia-pacific journal of operational research*, *33*(05).

[45] Ciavotta, M., Minella, G., & Ruiz, R. (2013). Multi-objective sequence dependent setup times permutation flowshop: A new algorithm and a comprehensive study. *Eur. J. of Operational Res. 227*(2), 301-313.

[46] Cochran, J. K., Horng, S. M., & Fowler, J. W. (2003). A multi-population genetic algorithm to solve multi-objective scheduling problems for parallel machines. *Comp. & Operations Res, 30*(7), 1087-1102.

[47] Coello, C. A. C., Lamont, G. B., & Van Veldhuizen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems* (Vol. 5). Springer-Verlag US.
[48] Crowder, B. (2006). *Minimizing the makespan in a flexible flowshop with sequence dependent setup times, uniform machines, and limited buffers*. West Virginia University.
[49] Dannenbring, D. G. (1977). An evaluation of flow shop sequencing heuristics. *Management science, 23*(11), 1174-1182.
[50] Davoudpour, H., & Ashrafi, M. (2009). Solving multi-objective SDST flexible flow shop using GRASP algorithm. *The International J. of Advanced Manufacturing Tech*, *44*(7), 737-747.
[51] Defersha, F. M. (2015). A simulated annealing with multiple-search paths and parallel computation for a comprehensive flowshop scheduling problem. *Int. Trans.in Operations Res. 22*(4), 669–691.
[52] De Weck, O. L. (2004). Multiobjective optimization: History and promise. *The third china-japan-korea joint symposium on optimization of structural and mechanical systems*. Kanazawa, Japan.
[53] Dhingra, A. K., & Chandna, P. (2015). Hybrid genetic algorithm for SDST flow shop scheduling with due dates: a case study. *Int. J. of Advanced Operations Management, 2*(3-4), 141-161.
[54] Du, X., Ji, M., Li, Z., & Liu, B. (2016). Scheduling of stochastic distributed assembly flowshop under complex constraints. *IEEE Symposium Series on Comp. Intel.* Athens, Greece.
[55] Ebrahimi, M., Fatemi Ghomi, S. M. T., & Karimi, B. (2014). Hybrid flow shop scheduling with sequence dependent family setup time and uncertain due dates. *Appl. Math. Modelling, 38*(9), 2490-2504.
[56] Eren, T. (2010). A bicriteria m-machine flowshop scheduling with sequence-dependent setup times. *Applied mathematical modelling, 34*(2), 284-293.
[57] Eskandari, H., & Hosseinzadeh, A. (2014.) A variable neighbourhood search for hybrid flow-shop scheduling problem with rework and set-up times. *Journal of the operational research society, 65*(8), 1221-1231.
[58] Fadaei, M., & Zandieh, M. (2013). Scheduling a bi-objective hybrid flow Shop with sequence-dependent family setup times using metaheuristics. *Arabian J. for Sci. & Eng, 38*(8), 2233-2244.
[59] Farahmand-Mehr, M., Fattahi, P., Kazemi, M., Zarei, H., & Piri, A. (2014). An efficient genetic algorithm for a hybrid flow shop scheduling problem with time lags and sequence-dependent setup time. *Manufacturing review, 1*(21), 1-10.
[60] Fattahi, P., Hosseini, S., & Jolai, F. (2013a). A mathematical model and extension algorithm for assembly flexible flow shop scheduling problem. *The Int. J. of Advanced Manuf. Tech*, *65*(5-8), 787–802.
[61] Fattahi, P., Hosseini, S., & Jolai, F. (2013b). Some heuristics for the hybrid flow shop scheduling problem with setup and assembly operations. *Int. J. of Industrial Eng. Computations, 4*(3), 393-416.
[62] Fattahi, P., Hosseini, S., Jolai, F., & Tavakkoli-Moghaddam, R. (2014). A branch and bound algorithm for hybrid flow shop scheduling problem with setup time and assembly operations. *Appl. Math. Mod, 38*(1),119-134.
[63] Frisch, A. M., Hnich, B., Kiziltan, Z., Miguel, I., & Walsh, T. (2006). Propagation algorithms for lexicographic ordering constraints. *Artificial intelligence, 170*(10), 803-834.
[64] Fu, Y., Wang, H., Huang, M., Ding, J., & Tian, G. (2017). Multiobjective flow shop deteriorating scheduling problem via an adaptive multipopulation genetic algorithm. *Proceedings of the institution of mechanical engineers part b journal of engineering manufacture*. SAGE Publishing.
[65] Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of np-completeness, computers and intractability: A guide to the theory of NP-Comp*. WH Freeman & Co. New York, USA.
[66] Ghafari, E., & Sahraeian, R. (2014). A two-stage hybrid flowshop scheduling problem with serial batching. *International journal of industrial engineering and production research, 25*(1), 55-63.
[67] Ghassemi-Tari, F., & Olfat, L. (2004). Two COVERT based algorithms for solving the generalized flow shop problems. *Proceedings of the 34th Int. Con. on Comp. and Industrial Eng*. San Francisco, CA, USA.

[68] Ghassemi-Tari, F., & Olfat, L. (2007). Development of a set of algorithms for the multi-project scheduling problems. *Journal of industrial and systems engineering, 1*(1), 11-17.
[69] Ghassemi-Tari, F., & Olfat, L. (2008). Covert based algorithms for solving the generalized tardiness flow shop problems. *Journal of industrial and systems engineering, 2*(3), 197-213.
[70] Ghassemi-Tari, F., & Olfat, L. (2010). A set of algorithms for solving the generalized tardiness flowshop problems. *Journal of industrial and systems engineering, 4*(3),156-166.
[71] Ghassemi-Tari, F. & Alaei, R. (2013). Scheduling TV commercials using genetic algorithms. *International journal of production research, 51*(16), 4921-4929.
[72] Ghassemi-Tari, F., & Olfat, L. (2014). Heuristic rules for tardiness problem in flow shop with intermediate due dates. *The Int. J. of Advanced Manuf. Tech, 71*(1-4), 381-393.
[73] Ghassemi Tari, F. & Hashemi Z. (2016). A priority based genetic algorithm for nonlinear transportation costs problems. *Computers and industrial engineering, 96*(c).
[74] Ghassemi-Tari, F., & Meshkinfam, S. (2017). Improving performance of GAs by use of selective breading evolutionary process. *British J. of Math. & Computer Sci, 22*(3), 1-21.
[75] Gicquel, C., Hege, L.,. Minoux, M., & Van, W. (2012). A discrete time exact solution approach for a complex hybrid flow-shop scheduling problem with limited-wait constraints. *Computers and operations research, 39*(3), 629-636.
[76] Glover, F. (1977). Heuristics for integer programming using surrogate constraints. *Dec. Sc, 8*(1), 156-166.
[77] Gholami, M., Zandieh, M., & Alem-Tabriz, A. (2009). Scheduling hybrid flow shop with sequence-dependent setup times and machines with random breakdowns. *The int. j. of ad. manuf. Tech, 42*(1-2), 189–201.
[78] Gholami S., & Rajaee Abyaneh F. (2016). Efficient algorithms for solving flexible flow shop scheduling problem with unrelated parallel machines and sequence-dependent setup times considering earliness/tardiness minimization.
[79] Gómez-Gasquet, P., Andrés, C., & Lario, F. C. (2012). An agent-based genetic algorithm for hybrid flowshops with sequence dependent setup times to minimise makespan. *Exp. Sys.with Appl, 39*(9), 8095-8107.
[80] Guinet, A. (1991) Textile production systems: a succession of non-identical parallel processor shops. *Journal of the operational research society,* 655-671.
[81] Gupta, J. N. (1971). A functional heuristic algorithm for the flowshop scheduling problem. *Journal of the operational research society, 22*(1), 39-47.
[82] Gupta J. N. (1979). A review of flowshop scheduling research. In Ritzman, L. P., Krajewski, L.J., Berry, W.L., Goodman, S.H., Hardy, S.T., & Vitt, L.D. (Eds), *Disaggregation*. Springer, Dordrecht, 363-388.
[83] Gupta, J. N., & Darrow, W. P. (1986). The two-machine sequence dependent flowshop scheduling problem. *European journal of operational research, 24*(3), 439-446.
[84] Gupta, D., Sharma, S., & Nailwal, K. K. (2012). A bicriteria two machine flowshop scheduling with sequence dependent setup time. *International journal of mathematical sciences, 11*(3-4), 183-196.
[85] Ha, B. B., & Duc, N. N. (2013). Multiple trajectory search for large scale global optimization search-based software testing: past, present and future Wolf search algorithm with ephemeral memory. *Int. Conf. on Comp. and Commun. Tech. Res. Innov. and Vision for the Future*. Hanoi, Vietnam.
[86] Haddad, M. N., Cota, L. P., Souza, M. J. F., & Maculan, N. (2014). AIV: A heuristic algorithm based on iterated local search and variable neighborhood descent for solving the unrelated parallel machine scheduling problem with setup times. *16th international conference on enterprise information systems*. Lisboa, Portugal.
[87] Hakimzadeh Abyaneh, S., & Zandieh, M. (2012). Bi-objective hybrid flow shop scheduling with sequence-dependent setup times and limited buffers. *Int J Adv Manuf Tech, 58*(1–4), 309–325.
[88] Harbaoui, H., Bellenguez-Morineau, O., & Khalfallah, S. (2016). Scheduling a two-stage hybrid flow shop with dedicated machines, time lags and sequence-dependent family setup times. *IEEE Int. Conference on Systems, Man, and Cybernetics (SMC).* Budapest, Hungary.

[89] Hashemi, Z., & Tari, F. G. (2018). A Prufer-based genetic algorithm for allocation of the vehicles in a discounted transportation cost system. *International journal of systems science: Operations & logistics*, *5*(1), 1-15.
[90] Hatami, S., Ebrahimnejad, S., Tavakkoli-Moghaddam, R., & Maboudian, Y. (2010). Two meta-heuristics for three-stage assembly flowshop scheduling with sequence-dependent setup times. *The international journal of advanced manufacturing technology, 50*(9), 1153-1164.
[91] Hatami, S., García, R. R., Romano, C. A. (2015). The distributed assembly parallel machine scheduling problem with eligibility constraints. *Int. J. of Prod. Manag. and Eng, 3*(1), 13-23.
[92] He, D.W., Kusiak, A., & Artiba A. (1996). A scheduling problem in glass manufacturing. *IIE transactions 28*(2), 129-139.
[93] Hecker, F. T., Hussein, W. B., Paquet-Durand, O., Hussein, M. A., & Becker, T. (2013). A case study on using evolutionary algorithms to optimize bakery production planning. *Expert systems with applications, 40* (17), 6837-6847.
[94] Hecker, F. T., Stanke, M., Becker, T., & Hitzmann, B. (2014). Application of a modified GA, ACO and a random search procedure to solve the production scheduling of a case study bakery. *Expert systems with applications, 41* (13), 5882-5891.
[95] Hekmatfar, M., Ghomi, S. M. T., & Karimi, B. (2011). Two stage reentrant hybrid flow shop with setup times and the criterion of minimizing makespan. *Applied soft computing, 11*(8), 4530-4539.
[96] Hendizadeh, S. H., ElMekkawy, T. Y., & Wang, G. G. (2007). Bi-criteria scheduling of a flowshop manufacturing cell with sequence dependent setup times. *European journal of industrial engineering, 1*(4), 391-413.
[97] Hidri, L., & Gharbi, A. (2017). New efficient lower bound for the Hybrid Flow Shop Scheduling Problem with Multiprocessor Tasks. *IEEE access, 5*, 6121-6133.
[98] Hosseini, S. M. H. (2016). Modeling the hybrid flow shop scheduling problem followed by an assembly stage considering aging effects and preventive maintenance activities. *International Journal of Supply and Operations management, 3*(1), 1215-1233.
[99] Javadian, N., Amiri-Aref, M., Hadighi, A., Kazemi, M., & Moradi, A. (2010). Flexible flow shop with sequence-dependent setup times and machine availability constraints. *International Journal of management science and engineering management, 5*(3), 219-226.
[100] Javadian, N., Fattahi, P., Farahmand-Mehr, M., Amiri-Aref, M., & Kazemi, M. (2012). An immune algorithm for hybrid flow shop scheduling problem with time lags and sequence-dependent setup times. *The international journal of advanced manufacturing technology, 63*(1), 337-348.
[101] Jin, Z., Ohno, K., Ito, T., & Elmaghraby, S. (2002). Scheduling hybrid flowshops in printed circuit board assembly lines. *Production and operations management, 11*(2), 216-230.
[102] Johnson, S. M. (1954) Optimal two-and three-stage production schedules with setup times included. *Naval research logistics (NRL), 1*(1), 61-68.
[103] Jolai, F., Sheikh, S., Rabbani, M., & Karimi, B. (2009). A genetic algorithm for solving no-wait flexible flow lines with due window and job rejection. *The Int. J. of Ad. Man. Tech, 42*(5), 523-532.
[104] Jolai, F., Rabiee, M., & Asefi, H. (2012). A novel hybrid meta-heuristic algorithm for a no-wait flexible flow shop scheduling problem with sequence dependent setup times. *Int. J. of Prod. Res. 50*(24), 7447-7466.
[105] Juan, A. A., Lourenço, H. R., Mateo, M., Luo, R., & Castella, Q. (2014). Using iterated local search for solving the flow-shop problem: Parallelization, parametrization, and randomization issues. *International transactions in operational research, 21*(1), 103-126.
[106] Jungwattanaki, J., Reodecha, M., Chaovalitwongse, P., & Werner, F. (2005). An evaluation of sequencing heuristics for flexible flowshop scheduling problems with unrelated parallel machines and dual criteria. *Otto-von-guericke-universitat magdeburg, 28*(05), 1-23.
[107] Jungwattanakit, J., Reodecha, M., Chaovalitwongse, P., & Werner, F. (2006). *Sequencing algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria*. INTAS

[108] Jungwattanakit, J., Reodecha, M., Chaovalitwongse, P., & Werner, F. (2008). Algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. *The international journal of advanced manufacturing technology, 37*(3-4), 354-370.
[109] Jungwattanakit, J., Reodecha, M., Chaovalitwongse, P., & Werner, F. (2009). A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. *Computers & operations research*, *36*(2), 358-378.
[110] Kangarloo, N., Rezaeian, J., & Khosrawi, X. (2016). JIT scheduling problem on flexible flow shop with machine break down, machine eligibility and setup times. *J. of Math. & Com. Sc*, *16*(1), 50-68.
[111] Karimi, N., Zandieh, M., & Karamooz, H. (2010). Bi-objective group scheduling in hybrid flexible flowshop: a multi-phase approach. *Expert systems with applications*, *37*(6), 4024-4032.
[112] Wang, H. (2005). Flexible flow shop scheduling: optimum, heuristics and artificial intelligence solutions. *Expert systems*, *22*(2), 78-85.
[113] Karmakar, S., & Mahanty, B. (2010). Minimizing makespan for a flexible flow shop scheduling problem in a paint company. *Preceding IEOM*. Dhaka, Bangladesh.
[114] Kayvanfar, V., Komaki, G. M., Aalaei, A., & Zandieh, M. (2014). Minimizing total tardiness and earliness on unrelated parallel machines with controllable processing times. *Comp. & Operations Res, 41*, 31-43.
[115] Khalili, M., & Tavakkoli-Moghaddam, R. (2012). A multi-objective electromagnetism algorithm for a bi-objective flowshop scheduling problem. *Journal of manufacturing systems, 31*(2), 232-239.
[116] Khalili, M., & Naderi, B. (2015). A bi-objective imperialist competitive algorithm for no-wait flexible flow lines with sequence dependent setup times. *The Int. J. of Ad. Man. Tech*, *76*(1-4), 461-469.
[117] Kia, H, Davoudpour, H, Zandieh, M. (2010). Scheduling a dynamic flexible flow line with sequence-dependent setup times: a simulation analysis. *Int. J. of Production Res. 48*(14): 4019-4042.
[118] Kia, H., Ghodsypour, SH., Davoudpour, H. (2017). New scheduling rules for a dynamic flexible flow line problem with sequence-dependent setup times. *J. of Industrial Eng. Int*. 1-10.
[119] Kianfar, K., Ghomi, S. M. T., & Jadid, A. O. (2012). Study of stochastic sequence-dependent flexible flow shop via developing a dispatching rule and a hybrid GA. *Eng. Appl. of Artificial Intel, 25*(3), 494-506.
[120] Komaki, M., Sheikh, S., Teymourian, E., & Malakooti, B. (2015). Cuckoo search algorithm for hybrid flow shop scheduling problem with multi-layer assembly operations. *International conference on operations excellence and service engineering*. Orlando, Florida, USA.
[121] Komaki, M., & Malakooti, B. (2017). General variable neighborhood search algorithm to minimize makespan of the distributed no-wait flow shop scheduling problem. *Prod. Eng. 11*(3), 1-15.
[122] Kumar, A., & Dhingra, A. (2010). Minimization of total weighted tardiness and makespan for SDST flow shop scheduling using genetic algorithm. *Int. J. of Applied Engineering Res, 3*(2), 483-494.
[123] Kurz, M. E. (2001). Scheduling flexible flow lines with sequence dependent setup Times. A Dissertation (Doctoral thesis, University of Arizona, USA).
[124] Kurz, M. E., & Askin, R. G. (2003). Comparing scheduling rules for flexible flow lines. *International journal of production economics, 85*(3), 371-388.
[125] Kurz, M. E., & Askin, R. G. (2004). Scheduling flexible flow lines with sequence-dependent setup times. *European journal of operational research, 159*(1), 66-82.
[126] Lee, S. M., & Asllani, A. A. (2004). Job scheduling with dual criteria and sequence-dependent setups: mathematical versus genetic programming. *Omega*, *32*(2), 145-153.
[127] Lee, G. C., Hong, J. M., & Choi, S. H. (2015). Efficient heuristic algorithm for scheduling two-stage hybrid flowshop with sequence-dependent setup times. *Mathematical Problems in Eng.*
[128] Lee, J. Y., & Bang, J. Y. (2016). A two-stage assembly-type flowshop scheduling problem for minimizing total tardiness. *Mathematical problems in engineering*.

[129] Li, L, Wang, L, Huo, J.(2010). Hybrid flowshop scheduling with setup times for cold treating process in Baoshan Iron & Steel Complex. *Int. Conference on Logistics Sys. and Intelligent Management*.

[130] Li, X., & Li, M. (2015). Multiobjective local search algorithm-based decomposition for multiobjective permutation flow shop scheduling problem. *IEEE Trans. on Eng. Manag, 62*(4), 544-557.

[131] Li, X., Ma, S (2016) Multi-objective memetic search algorithm for multi-objective permutation flow shop scheduling problem. *IEEE access, 4,* 2154-2165.

[132] Li, X, & Ma, S. (2017). Multiobjective discrete artificial bee colony algorithm for multiobjective permutation flow shop scheduling problem with sequence dependent setup times. *IEEE Trans. on Eng. Manag, 64*(2), 149-165.

[133] Lin, H. T., & Liao, C. J. (2003). A case study in a two-stage hybrid flow shop with setup time and dedicated machines. *International journal of production economics, 86*(2), 133-143.

[134] Lin, S. W., & Ying, K. C. (2012). Scheduling a bi-criteria flowshop manufacturing cell with sequence-dependent family setup times. *European journal of industrial engineering, 6*(4), 474-496.

[135] Liou, C. D., & Hsieh, Y. C. (2015). A hybrid algorithm for the multi-stage flow shop group scheduling with sequence-dependent setup and transportation times. *Int. J. of Prod. Eco, 170,* 258-267.

[136] Liu, C. Y., & Chang, S. C. (2000). Scheduling flexible flow shops with sequence-dependent setup effects. *IEEE transactions on robotics and automation, 16*(4), 408-419.

[137] Lu, D., & Logendran, R. (2013). Bi-criteria group scheduling with sequence-dependent setup time in a flow shop. *Journal of the operational research society, 64*(4), 530-546.

[138] Lu, C., Xiao, S., Li, X., & Gao, L. (2016). An effective multi-objective discrete grey wolf optimizer for a real-world scheduling problem in welding production. *Advances in Eng. Soft, 99,* 161-176.

[139] Lu, C., Gao, L., Li, X., Pan, Q.K., & Wang, Q. (2017a). Energy-efficient permutation flow shop scheduling problem using a hybrid multi-objective backtracking search algorithm. *J. of Cleaner Production, 144,* 228-238.

[140] Lu, C., Gao, L., Li, X., & Xiao, S. (2017b). A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry. *Eng. Appl.of Artificial Intelligence, 57,* 61-79.

[141] Luo, H., Huang, G. Q., Zhang, Y., Dai, Q., & Chen, X. (2009). Two-stage hybrid batching flowshop scheduling with blocking and machine availability constraints using genetic algorithm. *Robotics and computer-integrated manufacturing, 25*(6), 962-971.

[142] Luo, H., Zhang, A., & Huang, G. Q. (2015). Active scheduling for hybrid flowshop with family setup time and inconsistent family formation. *Journal of intelligent manufacturing, 26*(1), 169-187.

[143] Lv, Y., Zhang, J., & Qin, W. (2017.) A genetic regulatory network-based method for dynamic hybrid flow shop scheduling with uncertain processing times. *Applied sciences, 7*(1), 23-42.

[144] Maboudian, Y., & Shafaei, R. (2009). Modeling a bi-criteria two stage assembly flow shop scheduling problem with sequence dependent setup times. I*EEE international conference on industrial engineering and engineering management. Hong Kong, china.*

[145] Majazi Dalfard, V., Ardakani, A., & Banihashemi, T. N. S. (2011). Hybrid genetic algorithm for assembly flow-shop scheduling problem with sequence-dependent setup and transportation times. *The. Vjesnik, 18*(4), 497-504.

[146] Maleki-Darounkolaei, A., Modiri, M., Tavakkoli-Moghaddam, R., & Seyyedi, I. (2012). A three-stage assembly flow shop scheduling problem with blocking and sequence-dependent set up times. *Journal of industrial engineering international, 8*(1), 26.

[147] Maleki-Daronkolaei, A., & Seyedi, I. (2013). Taguchi method for three-stage assembly flow shop scheduling problem with blocking and sequence-dependent set up times. *J. of Eng. Sc.and Tech, 8*(5), 603-622.

[148] Mansouri, S. A., Hendizadeh, S. H., & Salmasi, N. (2009). Bicriteria scheduling of a two-machine flowshop with sequence-dependent setup times. *The Int. J. of Ad. Man.Tech, 40*(11), 1216-1226.

[149] Mastrolilli, M., & Svensson, O. (2011). Hardness of approximating flow and job shop scheduling problems. *Journal of the ACM (JACM), 58*(5),1-32.
[150] Mehravaran, Y., & Logendran, R. (2012). Non-permutation flowshop scheduling in a supply chain with sequence-dependent setup times. *Int. Journal of Production Economics, 135*(2), 953-963.
[151] Meshkinfam, S., Ghassemi, Tari F. (2016). A genetic algorithm for generalized tardiness flowshop problems. *International journal of engineering science and technology, 8*(9), 219-228.
[152] Michalewicz, Z. (1996). Heuristic methods for evolutionary computation techniques. *Journal of heuristics*, *1*(2), 177-206.
[153] Miettinen, K., Mäkelä, M. M., & Kaario, K. (2006). Experiments with classification-based scalarizing functions in interactive multiobjective optimization. *Eur. J. of Operational Res, 175*(2), 931-947.
[154] Miettinen, K., Molina, J., González, M., Hernández-Díaz, A., & Caballero, R. (2009). Using box indices in supporting comparison in multiobjective optimization. *Eur. J. of Operational Res, 197*(1), 17-24.
[155] Minella, G., Ruiz, R., & Ciavotta, M. (2011). Restarted iterated pareto greedy algorithm for multi-objective flowshop scheduling problems. *Computers & operations research*, *38*(11), 1521-1533.
[156] Mirsanei, H., Zandieh, M., Moayed, M. J., & Khabbazi, M. R. (2011). A simulated annealing algorithm approach to hybrid flow shop scheduling with sequence-dependent setup times. *J. of Int. Manuf, 22*(6), 965-978.
[157] Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Com. & Op. Res, 24*(11), 1097-1100.
[158] Hansen, P., Oğuz, C., & Mladenović, N. (2008). Variable neighborhood search for minimum cost berth allocation. *European journal of operational research, 191*(3), 636-649.
[159] Mohammadi, G. (2015). Multi-objective flow shop production scheduling via robust genetic algorithms optimization technique. *Int. J. of Service Sci., Manag. & Eng, 2*(1), 1-8.
[160] Mousavi, S., Zandieh, M., & Amiri, M. (2011). An efficient bi-objective heuristic for scheduling of hybrid flow shops. *The Int. Journal of Advanced Manufacturing Technology, 54*(1), 287-307.
[161] Mousavi, S., Zandieh, M., & Amiri, M. (2012). Comparisons of bi-objective genetic algorithms for hybrid flowshop scheduling with sequence-dependent setup times. *Int. J. of Prod. Res, 50*(10), 2570-2591.
[162] Mousavi, S., & Zandieh, M. (2016). An Efficient Hybrid Algorithm for a Bi-objectives Hybrid Flow Shop Scheduling. *Intelligent automation & soft computing*, 1-8.
[163] Mousavi, S., Mahdavi, I., Rezaeian, J., & Zandieh, M. (2016). An efficient bi-objective algorithm to solve re-entrant hybrid flow shop scheduling with learning effect and setup times. *Operational research*, 1-36.
[164] Mousavi, S.M., Mahdavi, I., Rezaeian J., & Zandieh, M. (2018). Bi-objective scheduling for the re-entrant hybrid flow shop with learning effect and setup times. *Scientia Iranica E, 25*(4), 2233{2253.
[165] Mozdgir, A., Fatemi Ghomi, S. M. T., Jolai, F., & Navaei, J. (2013). Two-stage assembly flow-shop scheduling problem with non-identical assembly machines considering setup times. *Int. J. of Prod. Res, 51*(12), 3625-3642.
[166] Naderi, B., Khalili, M., Taghavifard, M., & Roshanaei, V. (2008). A variable neighborhood search for hybrid flexible flowshops with setup times minimizing total completion time. *J. of Appl. Sci, 8*(16), 2843-2850.
[167] Naderi, B., Zandieh, M., & Fatemi Ghomi, S. M. T. (2009d). A study on integrating sequence dependent setup time flexible flow lines and preventive maintenance scheduling. *J. of Intelligent Manuf, 20*(6), 683-694.
[168] Naderi, B., Zandieh, M., Fatemi Ghomi, S. M. T. (2009c). An iterated greedy algorithm for flexible flow lines with sequence dependent setup times to minimize total weighted completion time. *Journal of optimization in industrial engineering, 2*(3), 33-37.
[169] Naderi, B., Zandieh, M., Balagh, A. K. G., & Roshanaei, V. (2009b). An improved simulated annealing for hybrid flowshops with sequence-dependent setup and transportation times to

minimize total completion time and total tardiness. *Expert systems with applications, 36*(6), 9625-9633.
[170] Naderi, B., Zandieh, M., & Roshanaei, V. (2009a). Scheduling hybrid flowshops with sequence dependent setup times to minimize makespan and maximum tardiness. *The Int. J. of Ad. Manf.Tech, 41*(11-12), 1186-1198.
[171] Naderi, B., Ruiz, R., & Zandieh, M. (2010). Algorithms for a realistic variant of flowshop scheduling. *Computers & operations research, 37*(2), 236-246.
[172] Naderi, B., Gohari, S., & Yazdani, M. (2014). Hybrid flexible flowshop problems: Models and solution methods. *Applied mathematical modelling, 38*(24), 5767-5780.
[173] Nawaz, M., Enscore, E. E., & Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega, 11*(1), 91-95.
[174] Nayak A., Fang K., & Lee S. (2017). Demand response in flow shop with job due dates using genetic algorithm approach. Smart and Sustainable Manufacturing Systems 1(1), 100-120.
[175] Nejati, M., Mahdavi, I., Hassanzadeh, R., & Mahdavi-Amiri, N. (2016). Lot streaming in a two-stage assembly hybrid flow shop scheduling problem with a work shift constraint. *J. of Ind. & Prod. Eng, 33*(7), 459-471.
[176] Pan, Q. K., Gao, L., Li X. Y., & Gao, K. Z. (2017). Effective metaheuristics for scheduling a hybrid flowshop with sequence-dependent setup times. *Appl. Mathematics and Comp, 303*, 89-112.
[177] Panahi, H. (2014). *Two-stage flexible flow shop scheduling with blocking constraint and batching machines* (Doctoral thesis, Oregon State University, Oregon, USA).
[178] Palmer, D. (1965). Sequencing jobs through a multi-stage process in the minimum total time-a quick method of obtaining a near optimum. *J. of the Operational Res. Society, 16*(1), 101-107.
[179] Patnaik, S., Yang, X. S., & Nakamatsu, K. (2017). *Nature-inspired computing and optimization: theory and applications, Vol. 10.* Springer International Publishing.
[180] Pargar, F., & Zandieh, M. (2012). Bi-criteria SDST hybrid flow shop scheduling with learning effect of setup times: water flow-like algorithm approach. *Int. J. of Prod. Res, 50*(10), 2609-2623.
[181] Parveen, S., & Ullah, H. (2011). Review on job-shop and flow-shop scheduling using multi criteria decision making. *Journal of mechanical engineering, 41*(2), 130-146.
[182] Pearn, W., Chung, S., Yang, M., & Chen, C. (2005). The integrated circuit packaging scheduling problem (icpsp): A case study. *Int. J. of Industrial Eng.-Theory Appl. and Practice 12*(3), 296-307.
[183] Perez-Gonzalez, P., & Framinan, F. M. (2015). Assessing scheduling policies in a permutation flowshop with common due dates. *International journal of production research, 53*(19), 5742-5754.
[184] Ponnambalam, S., Jagannathan, H., Kataria, M., & Gadicherla, A. (2004) A TSP-GA multi-objective algorithm for flow-shop scheduling. *The Int J. of Advanced Manuf. Tech, 23*(11-12), 909-915.
[185] Qian, B., Wang, L., Huang, D. X., Wang, W. I., & Wang, X. (2009) An effective hybrid DE-based algorithm for multi-objective flow shop scheduling with limited buffers. *Comp. & Op. Res, 36*(1), 209-233.
[186] Rabiee, M., Rad, R. S., Mazinani, M., & Shafaei, R. (2014). An intelligent hybrid meta-heuristic for solving a case of no-wait two-stage flexible flow shop scheduling problem with unrelated parallel machines. *The Int. J. of Ad. Man. Tech, 71*(5-8), 1229-1245.
[187] Rahimi-Vahed, A., & Mirzaei, A. H. (2007). A hybrid multi-objective shuffled frog-leaping algorithm for a mixed-model assembly line sequencing problem. *Comp. & Industrial Eng, 53*(4), 642-666.
[188] Rahmanidoust, M., Zheng, J., & Rabiee, M. (2017). Simultaneous considering of machine availability constraint, sequence dependent setup time and ready time in a no-wait hybrid flow shop scheduling problem to minimize mean tardiness. *International journal of computer applications, 169*(7), 30-37.
[189] Rajaee Abyaneh, F., & Gholami, S. (2015). A comparison of algorithms for minimizing the sum of earliness and tardiness in hybrid flow-shop scheduling problem with unrelated parallel machines and sequence-dependent setup times. *Industrial and systems engineering, 8*(2), 67-85.

[190] Ramezani, P., Rabiee, M., & Jolai, F. (2015). No-wait flexible flowshop with uniform parallel machines and sequence-dependent setup time: a hybrid meta-heuristic approach. *J. of Int. Man, 26*(4), 731-744.
[191] Ramezanian, R., Aryanezhad, M., & Heydari M. (2010). A mathematical programming model for flow shop scheduling problems for considering just in time production. *Int. J. of Industrial Eng, 21*(2), 97-104.
[192] Ramezanian, R., Fallah Sanami, S., & Mahmoodian, V. (2017). A new mathematical model for integrated production planning and scheduling problem in capacitated flexible flow shop with sequence-dependent setups. *Scientia iranica, 24*(5), 2501-2514.
[193] Rashidi, E., Jahandar, M., & Zandieh, M. (2010). An improved hybrid multi-objective parallel genetic algorithm for hybrid flow shop scheduling with unrelated parallel machines. *The Int. J. of Ad. Manuf. Tech, 49*(9), 1129-1139.
[194] Rezaeian, J., Seidgar, H., & Kiani, M. (2013). Scheduling of a flexible flow shop with multiprocessor task by a hybrid approach based on genetic and imperialist competitive algorithms. *J. of Opt. in Ind. Eng, 6*(13), 1-11.
[195] Ribas, I., Companys, R., & Tort-Martorell, X. (2017). Efficient heuristics for the parallel blocking flow shop scheduling problem. *Expert systems with applications, 74*, 41-54.
[196] Ruiz, R., & Maroto, C. (2006). A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. *European journal of operational research, 169*(3), 781-800.
[197] Ruiz, R., Şerifoğlu, F. S., & Urlings, T. (2008). Modeling realistic hybrid flexible flowshop scheduling problems. *Computers & operations research, 35*(4), 1151-1175.
[198] Saluja, V., & Jain, A. (2014). Optimization of flexible flow shop scheduling with sequence dependent setup time and lot splitting. *5th international & 26th all india manufacturing technology, design and research conference*. IIT Guwahati, Assam, India
[199] Samarghandi, H., & Behroozi, M. (2017). On the exact solution of the no-wait flow shop problem with due date constraints. *Computers & operations research, 81*, 141-159.
[200] Saravanan, M., Vijayakumar, S. J. D., & Srinivasan, R. (2014). A multicriteria permutation flowshop scheduling problem with setup times. *Int. J. of Eng. and Tech, 6*(3), 1329-1339.
[201] Satyanarayana, D., & Pramiladevi, M. (2016). Multi-criteria m-machine SDST flow shop scheduling using modified heuristic genetic algorithm. *Int. J. of Industrial and Sys. Eng, 22*(4), 409-422.
[202] Satyanarayana, D., & Pramiladevi, M. (2017). Special heuristics for flowshop scheduling based on hybrid genetic algorithm under SDST environment. *Int. J. of Mech. Eng. & Tech, 8*(4), 327-336.
[203] Seidgar, H., Ezzati, M., Kiani, M., & Tavakkoli-Moghaddam, R. (2013). An efficient genetic algorithm for two-stage hybrid flow shop scheduling with preemption and sequence dependent setup time. *Journal of mathematics and computer science, 6*, 251 – 259.
[204] Seidgar, H., Abedi, M., & Rad, S. T. (2015). A new mathematical model for scheduling flexible flow shop problem with learning and forgetting effects of workers. *Int. J.of Industrial and Sys. Eng, 21*(4), 534-549.
[205] Seyedi, I., Maleki-Daronkolaei, A., & Kalashi, F. (2012). Tabu search and simulated annealing for new three-stage assembly flow shop scheduling with blocking. *Int. J. of Cont. Res. In Business, 4*(8), 394-402.
[206] Shabtay, D., & Oron, D. (2016). Proportionate flow-shop scheduling with rejection. *Journal of the operational research society, 67*(5), 752-769.
[207] Shahvari, O., Salmasi, N., & Logendran, R. (2009). A meta-heuristic algorithm for flexible flow shop sequence dependent group scheduling problem. *Proceedings of the 2009 Int. Conf. on Value Chain Sustainability*. Kentucky, USA.
[208] Sharma, S., Gupta, D., & Nailwal, K. K. (2017a). Bi-criteria multistage flow shop scheduling with sequence-dependent setup times. *International journal of operational research, 29*(1), 127-147.
[209] Sharma, S., Gupta, D., & Nailwal, K. K. (2017b). Sequence dependent flow shop scheduling with job block criteria. *Global journal of pure and applied mathematics, 13*(5), 1401-1414.

[210] Sheikh, S., Komaki, M., Teymourian, E., & Malakooti, B. (2015). Multi-objective non-permutation flowshop with dependent setup times and missing operations. *7th Int. Joint Conf. on Comp. Intel.* (IJCCI).

[211] Sioud, A., Gagné, C., & Gravel, M. (2014a). Metaheuristics for solving a hybrid flexible flowshop problem with sequence-dependent setup times. *1st international conference on swarm intelligence based optimization, first international conference, ICSIBO*. Mulhouse, France.

[212] Sioud, A., Gagné, C., & Gravel, M. (2014). Minimizing total tardiness in a hybrid flexible flowshop with sequence dependent setup times. *The fourth international conference on advanced communications and computation*. Paris, France.

[213] Sioud, A., Gagné, C., & Dort, J. (2015). A GISMOO algorithm for a multi-objective permutation flowshop with sequence-dependent setup times. *7th international joint conference on computational intelligence*. Lisbon, Portugal.

[214] Sivapragasam, S., & Suppiah, Y. (2017). Minimizing Total Weighted Tardiness in Identical Parallel Machine with Sequence Dependent Setup Time Using Genetic Algorithm. *Journal of telecommunication, electronic and computer engineering (JTEC), 9*(1-4), 89-93.

[215] Song, J., Tang, J., Luo, X., & Liu, S. (2008). Scheduling model and heuristic algorithm for roller annealing. *Chinese controland decision conference, CCDC* (pp. 1052-1055).

[216] Sukkerd, W., & Wuttipornpun, T. (2017). Non-population search algorithms for capacitated material Requirement Planning in multi-stage assembly flow shop with alternative machines. *World academy of science, engineering and technology, Int. J. of Mech., Aer., Ind., Mech. and Man. Eng,* 11(3), 636-643.

[217] Tabrizi, A. A., Zandieh, M., & Vaziri, Z. (2009). A novel simulated annealing algorithm to hybrid flow shops scheduling with sequence-dependent setup times. *J. of Applied Sciences, 9*(10), 1943-1949.

[218] Tang, J., & Song, J. (2010). Discrete particle swarm optimization combined with no-wait algorithm in stages for scheduling mill roller annealing process. *International journal of computer integrated manufacturing, 23* (11), 979-991.

[219] Tao, W., & Rongqiu, C. (2006). Heuristic algorithm of scheduling model for cold coils annealing. *Proceedings of the world congress on intelligent control and automation (WCICA)* (pp. 7263-7266).

[220] Tavakkoli-Moghaddam R., & Safaei, N. (2007). A new mathematical model for flexible flow lines with blocking processor and sequence-dependent setup time. *Multiprocessor scheduling, theory and applications: I-Tech education and publishing*. Vienna, Austria 2007, 266–283.

[221] Tavakkoli-Moghaddam, R., Rahimi-Vahed, A., & Mirzaei, A.H. (2007). A hybrid multi-objective immune algorithm for a flow shop scheduling problem with bi-objectives: weighted mean completion time and weighted mean tardiness. *Information sciences, 177*(22), 5072-5090.

[222] Tavakkoli-Moghaddam, R., Azarkish, M., & Sadeghnejad-Barkousaraie, A. (2011). A new hybrid multi-objective Pareto archive PSO algorithm for a bi-objective job shop scheduling problem. *Expert Sys. with Appl*, *38*(9), 10812-10821.

[223] Tavakkoli-Moghaddam, R., & Amin-Tahmasbi, H. (2012). A multi-objective immune system for a new bi-objective permutation flowshop problem with sequence-dependent setup times. *Iranian journal of operations research, 3*(2), 66-82.

[224] Taillard, E. (1993) Benchmarks for basic scheduling problems. *European journal of operational research*, *64*(2), 278-285.

[225] Tian, H., Li, K., & Liu, W. (2016). A Pareto-based adaptive variable neighborhood search for biobjective hybrid flow shop scheduling problem with sequence-dependent setup time. *Math. Pro. in Eng*.

[226] Tyaghi, N., Seidgar, H., Abedi, M., & Chandramouli, A. B. (2014). Learning and forgetting effects of flexible flow shop scheduling. *International journal of innovation and applied studies, 7*(3), 857-867.

[227] Urlings, T., Ruiz, R., & Serifoglu, F. S. (2010). Genetic algorithms with different representation schemes for complex hybrid flexible flow line problems. *International journal of metaheuristics, 1*(1), 30-54.

[228] Vanchipura, R., Sridharan, R., & Babu, A. S. (2014). Improvement of constructive heuristics using variable neighbourhood descent for scheduling a flow shop with sequence dependent setup time. *Journal of manufacturing systems, 33*(1), 65-75.
[229] Wang, H. S., Wang, L. C., Chen, T. L., Chen, Y. Y., & Cheng, C. Y. (2013a). Multi-stage parallel machines and lot-streaming scheduling problems–a case study for solar cell industry. In Prabhu, V., Taisch, M., & Kiritsis D (Eds). *Advances in production management systems* (pp. 151-158). Springer, Berlin, Heidelberg.
[230] Wang, I. L., Wang, Y. C., & Chen, C. W. (2013b). Scheduling unrelated parallel machines in semiconductor manufacturing by problem reduction and local search heuristics. *Flex. Ser. & Man. J, 25*(3), 343-366.
[231] Xiao, J., & Zheng, L. (2010). A MILP-based batch scheduling for two-stage hybrid flowshop with sequence-dependent setups in semiconductor assembly and test manufacturing. *IEEE conference on automation science and engineering* (pp. 21-24). Toronto, ON, Canada.
[232] Xu, J. Y., Dong, N. Q., & Gu, S. S. (2013). Multi-objective permutation flowshop scheduling with sequence-dependent setup times. *Computer integrated manufacturing systems, 19*(12).
[233] Xu, Z. H., Li, J. M., & Gu, X. S. (2016). Multi-objective flow shop scheduling problem based on GMOGSO. *Control and decision, 31*(10), 1772-1778.
[234] Xu, J., Wu, C. C., Yin, Y., & Lin, W. C. (2017). An iterated local search for the multi-objective permutation flowshop scheduling problem with sequence-dependent setup times. *Appl. Soft Comp, 52*, 39-47.
[235] Yaurima, V., Burtseva, L., & Tchernykh, A. (2009). Hybrid flowshop with unrelated machines, sequence-dependent setup time, availability constraints and limited buffers. *Comp. & Ind. Eng, 56*(4), 1452-1463.
[236] Yazdani, M., & Naderi, B. (2017). Modeling and scheduling no-idle hybrid flow shop problems. *J. of Opt. in Industrial Eng, 10*(21), 59-66.
[237] Yokoyama, M. (2001). Hybrid flow-shop scheduling with assembly operations. *International journal of production economics, 73*(2), 103-116.
[238] Zandieh, M., Fatemi Ghomi, S. M. T., & Husseini, S. M. (2006). An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times. *Appl. Math. and Comp, 180*(1), 111-127.
[239] Zandieh, M., & Rashidi, E. (2009). An effective hybrid genetic algorithm for hybrid flow shops with sequence dependent setup times and processor blocking. *J. of Industrial Eng, 4*(8), 51-58.
[240] Zandieh, M., & Gholami, M. (2009). An immune algorithm for scheduling a hybrid flow shop with sequence-dependent setup times and machines with random breakdowns. *Int. J. of Prod. Res, 47*(24), 6999-7027.
[241] Zandieh, M., Dorri, B., & Khamseh, A. (2009). Robust metaheuristics for group scheduling with sequence-dependent setup times in hybrid flexible flow shops. *The Int. J.of Adv. Manuf. Tech, 43*(7), 767-778.
[242] Zandieh, M., Mozaffari, E., & Gholami, M. (2010). A robust genetic algorithm for scheduling realistic hybrid flexible flow line problems. *Journal of intelligent manufacturing, 21*(6), 731-743.
[243] Zandieh, M., & Karimi, N. (2011). An adaptive multi-population genetic algorithm to solve the multi-objective group scheduling problem in hybrid flexible flowshop with sequence-dependent setup times. *Journal of intelligent manufacturing, 22*(6), 979-989.
[244] Zandieh, M., & Hashemi, A. (2015). Group scheduling in hybrid flexible flowshop with sequence-dependent setup times and random breakdowns via integrating genetic algorithm and simulation. *International journal of industrial and systems engineering, 21*(3), 377-394.