The hybrid flow shop scheduling problem

Article *in* European Journal of Operational Research · August 2010

DOI: 10.1016/j.ejor.2009.09.024 · Source: RePEc

Rubén Ruiz
Universitat Politècnica de València

José Antonio Vázquez Rodríguez

The Hybrid Flow Shop Scheduling Problem

Rubén Ruiz[1], José Antonio Vázquez-Rodríguez[2],

[1] Grupo de Sistemas de Optimización Aplicada, Instituto Tecnológico de Informática,

Universidad Politécnica de Valencia, Valencia, Spain. rruiz@eio.upv.es

[2]Automated Scheduling, Optimisation and Planning research group

School of Computer Science, University of Nottingham

Jubilee Campus Wollaton Road, Nottingham, NG8 1BB, U.K. jav@cs.nott.ac.uk

March 1, 2009

Abstract

The scheduling of flow shops with multiple parallel machines per stage, usually referred to as the Hybrid Flow Shop (HFS), is a complex combinatorial problem encountered in many real world applications. Given its importance and complexity, the HFS problem has been intensively studied. This paper presents a literature review on exact, heuristic and metaheuristic methods that have been proposed for its solution. The paper discusses several variants of the HFS problem, each in turn considering different assumptions, constraints and objective functions. Research opportunities in HFS are also discussed.

1 Introduction

Hybrid Flow Shops (HFS) are common manufacturing environments in which a set of n jobs are to be processed in a series of m stages. There are a number of variants, all of which have most of the following characteristics in common:

1. The number of processing stages m is at least 2,

2. Each stage k has $M^{(k)} \geq 1$ machines in parallel and in at least one of the stages $M^{(k)} > 1$,

3. 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 problem is to find a schedule which optimizes a given objective function. The HFS problem is, in most cases, $\mathcal{NP}$-hard. For instance, HFS restricted to two processing stages, even in the case when one stage contains two machines and the other one a single machine, is $\mathcal{NP}$-hard, after the results of [62]. Similarly, the HFS when machines are allowed to stop processing operations before their completion and to resume them on different time slots (something referred to as preemption) results also in strongly $\mathcal{NP}$-hard problems even with $m = 2$, according to [78]. Moreover, the special case where there is a single machine per stage, known as the flow shop, and the case where there is a single stage with several machines, known as the parallel machines environment, are also $\mathcal{NP}$-hard, [51]. However, with some special properties and precedence relationships, the problem might be polynomially solvable ([45]).

HFS is found in all kinds of real world scenarios including the electronics [217], [218], [118], [87], paper, [177] and textile, [55], industries. Examples are also found in the production of concrete, [143], the manufacturing of photographic film, [197, 4], and others, [2, 42, 3, 116, 20, 224, 159]. We also find examples in non-manufacturing areas like civil engineering [46], internet service architectures [8] and container handling systems [38, 37].

The HFS problem has attracted a lot of attention given its complexity and practical relevance. This paper describes the HFS problem and reviews many of the solution approaches that have been proposed for its solution. These include exact methods, heuristics, and metaheuristics. The present review fills in some of the gaps identified in previous reviews, like those of, [204, 117, 213] or more recently [151], and describes the most recent approaches. It also identifies research opportunities and proposes some interesting research lines.

The rest of the paper is organized as follows. A formal description of the HFS problem is given in Section 2. Section 3 explains the terminology used to refer to the different assumptions, constraints and objective functions, which combined, lead to different HFS scenarios. An extensive literature review is given in Section 4. This literature is carefully analyzed in Section 5. Finally, Section 6 discusses some research opportunities in HFS and concludes the paper.

2 Problem description

This section describes the HFS problem in its "standard" form using a mathematical programming formulation similar to that in [24]. In the standard problem all jobs and machines are available at time zero, machines at a given stage are identical, any machine can process only one operation at a time and any job can be processed by only one machine at a time; setup times are negligible, preemption is not allowed, the capacity of buffers between stages is unlimited and problem data is deterministic and know in advance. Although most of the problems described in the forthcoming sections do not fully complain with these assumptions, they mostly differ in two or three aspects only; the standard problem will serve as a "template" to which assumptions and constraints will be added or removed to describe different HFS variants.

In what follows, let j be the index which identifies a job, k a stage, and l the l^{th} machine of a given stage. Every job requires a set of operations to be performed sequentially; denote by p_{jk} the processing time required by job j in stage k. Given a schedule, let c_{jk} be the completion time of job j in stage k and let $c_{j0} = 0$.

Let

$$Y_{jkl} = \begin{cases} 1 \text{ if job } j \text{ on stage } k \text{ is scheduled in machine } l \\ 0 \text{ otherwise} \end{cases},$$

$$X_{jrk} = \begin{cases} 1 \text{ if job } j \text{ precedes job } r \text{ on stage } k \\ 0 \text{ otherwise} \end{cases},$$

and $Q \geq \sum_j \sum_k M^{(k)} p_{jk}$ be an arbitrarily large number. Let Z be the objective function, the model is

$$\text{minimize } Z \tag{1}$$

$$s.t.$$

$$Z \geq c_{jm}, \forall j \tag{2}$$

$$\sum_{l=1}^{M^{(k)}} Y_{jkl} = 1, \forall (j,k) \tag{3}$$

$$c_{jk} - c_{j,k-1} \geq \sum_{l=1}^{M^{(k)}} Y_{jkl} p_{jk}, \forall (j,k) \tag{4}$$

$$Q(2 - Y_{jkl} - Y_{qkl} + X_{jqk}) + c_{jk} - c_{qk} \geq p_{jk}, \forall (j,k,l,q) \text{ such that } j < r \tag{5}$$

$$Q(3 - Y_{jkl} - Y_{qkl} - X_{jqk}) + c_{qk} - c_{jk} \geq p_{qk}, \forall (j,k,l,q) \text{ such that } j < r \tag{6}$$

$$Y_{jkl} \in \{0,1\}, \forall (j,k,l) \tag{7}$$

$$X_{jrk} \in \{0,1\}, \forall (j,q,k) \tag{8}$$

$$c_{jk} \geq 0, \forall (j,k). \tag{9}$$

Z is restricted to be greater or equal to the completion time of the last operation to finish its processing, i.e., the makespan. Several other criteria will be discussed in Section 3. The set of constraints (3), guarantees that all operations are assigned strictly to one machine at each stage. Constraint set (4), restrict the starting time of operation o_{jk} to be greater or equal to its release time from the previous stage. Constraint sets (5) and (6) prevent any two operations from overlapping in a common machine. Constraint sets (7), (8) and (9) define the domains of the decision variables.

The HFS problem can also be represented as a graph $G(N, A)$, where N is a set of nodes corresponding to each operation, and A is a set of disjunctive arcs describing the set of possible paths in the graph. A solution is a graph $G(N, S)$, where S is a subset of the arcs in A but with a fixed direction, i.e., S represents an assignment and ordering of the job operations. Several heuristics have been devised using these representation, these will be discussed in Section 4.

3 Naming hybrid flowshop variants

The modification, removal, or addition of assumptions and/or constraints to the standard problem described above leads to different HFS variants. To refer to them, the nomenclature presented in [56] and further extended for the HFS case in [204], is adopted. Scheduling problems are described with a triplet $\alpha|\beta|\gamma$, where α describes a shop configuration, β a set of constraints and assumptions others than the ones listed in Section 2, and γ the objective function considered.

Parameter α defines the structure of the shop, including the number of stages and the number and characteristics of the machines per stage. α is composed of four parameters α_1, α_2, α_3 and α_4. α_1 indicates the general configuration of the shop, in this case a hybrid flowshop, denoted FH. α_2 is the number of stages in the shop. α_3 and α_4, together, describe the properties of the machines per stage. The notation $(\alpha_3\alpha_4)^k$ means that there are α_4 parallel machines of the type α_3 in stage k. $\alpha_3 \in \{\varnothing, P, Q, R\}$, where P indicates identical parallel machines, Q uniform parallel machines and R unrelated parallel machines, see [147] for definitions. In the case that there is a single machine, $\alpha_3 = \varnothing$.

The second element, β, lists the constraints and assumptions, other than those of the standard problem, which characterize the problem. The most common are:

- r_j indicates that job j cannot start processing before its release date r_j.
- $prmu$ indicates that the jobs are processed in every stage in the same order.
- $prec$ indicates that there are precedence constraints between operations from different jobs.
- M_j indicates that the processing of job j is restricted to the set of machines M_j at stage k. This is known as eligibility.
- S_{sd} indicates that the setup times are dependent on the sequence of operations.
- $prmp$ indicates that preemptions are permitted.
- $block$ implies that the buffer capacities between stages are limited. The jobs must wait in the previous stage until sufficient space is released.
- $recrc$ indicates that jobs are allowed/required to be processed more than once in the same stage.
- $unavail$ indicates that machines are not available at all times,
- $no - wait$ jobs are not allowed to wait between two successive stages. This implies that the shop operates under the First In First Out (FIFO) discipline.
- $p_j = p$ indicates that all processing times are equal to p.
- $size_{jk}$ indicates that o_{jk} must be processed on $size_{jk}$ machines simultaneously.

Table 1: Common objective functions

Notation	Description	Meaning
$C_{\max}$	$\max_j C_j$	maximum completion time
$F_{\max}$	$\max_j(C_j - r_j)$	maximum flow time
$L_{\max}$	$\max_j(L_j)$	maximum lateness
$T_{\max}$	$\max_j(T_j)$	maximum tardiness
$E_{\max}$	$\max_j(E_j)$	maximum earliness
$\bar{C}$	$\sum C_j$	total/average completion time
$\bar{C}^w$	$\sum w_j C_j$	total/average weighted completion time
$\bar{F}$	$\sum F_j$	total/average flow time
$\bar{F}^w$	$\sum w_j F_j$	total/average weighted flow time
$\bar{T}$	$\sum T_j$	total/average tardiness
$\bar{T}^w$	$\sum w_j T_j$	total/average weighted tardiness
$\bar{U}$	$\sum U_j$	number of late jobs
$\bar{U}^w$	$\sum w_j U_j$	total weighted number of late jobs
$\bar{E}$	$\sum E_j$	total/average earliness
$\bar{E}^w$	$\sum w_j E_j$	total/average weighted earliness

Let $C_j = c_{jm}$ be the completion time of job j at stage m. The flowtime of job j, F_j, is the time it spends in the system, i.e., $F_j = C_j - r_j$. The lateness of job j, or L_j, is $C_j - d_j$. Note that L_j may be a negative value. The tardiness, $T_j = \max\{C_j - d_j, 0\}$, and earliness, $E_j = \max\{d_j - C_j, 0\}$ are non-negative. $U_j \in \{1, 0\}$ is a penalty measure for every tardy job. It takes a value of 1 if $C_j - d_j > 0$ and 0 otherwise. It is common to associate a weight ω_j with job j in order to model its importance. Such a weight may represent, for instance, the cost of the job, or its volume, or relative priority, among other possibilities.

Elements C_j, F_j, etc. and their weighted counterparts $\omega_j C_j$, $\omega_j F_j$, etc. are frequently used to describe objective functions. See some common possibilities and their standard notation in Table 1. Functions involving the completion times of jobs (C_j) are concerned with the capacity utilization, i.e., the adequate exploitation of machinery. Due dates related functions, T, L and U, on the other hand, penalize failures to meet the clients demands on time. Functions involving E are usually associated with costs of inventory of finished products.

The standard problem, described in Section 2 is referred to as $FHm, ((PM^{(k)})_{k=1}^{m})||C_{\max}$, where FH_m means that the problem is a HFS with any number of stages m; $(PM^{(k)})_{k=1}^{m}$ means that in all stages, $k = 1, \ldots, m$, there are any number of identical parallel machines. The objective to optimize is the makespan, i.e., $C_{\max}$. Suppose that the number of stages is limited to 3, that there is only one machine in stage 1 and that there are any number of unrelated machines in stages 2 and 3. Moreover, jobs have release times, and setup times in stage 2 are sequence dependent. The objective to optimize is the sum of the weighted tardiness. This modified example problem would be referred to as $FH3, (1^{(1)}, ((RM^{(k)})_{k=2}^{3})|r_j, S_{sd}^{(2)}|\overline{T}^w$.

4 Literature review

The described nomenclature was used to summarize the type of problem addressed in more than 200 papers. This information is presented in Table 3 in the Appendix, where the first column indicates the year of publication, the second is the bibliographical reference, the third describes the problem addressed in the paper and the final column briefly describes the type of approach proposed as well as other details that may be of interest. Table 3 complements and follows a similar format to those presented in [159] and [204].

Notice that there are very many variants of the HFS scheduling problem. Some variants deviate enough from what was defined as the standard HFS problem as to be considered separately. For example, Flexible Manufacturing Systems (FMS) include, but are not limited to, HFS. In this paper, we stay within a reasonable scope and only consider problems that, according to the authors' discretion, are either special cases of the standard problem, or more general cases that are the result of the addition or removal of a limited number of assumptions and/or constraints. For a more in depth review about FMS, the reader is referred to [138], [152], [194] or [17] to cite just a few.

The rest of this section describes approaches to the different variants of the problem. Given the large number of approaches and problem variants we opted for a simple classification with the three very broad classes: exact algorithms, deterministic heuristics and metaheuristics, as we believe it to be more appropriate than other more sophisticated classifications which after all could not capture the wide variety of the HFS literature.

4.1 Exact algorithms

Without doubt, Branch and Bound (B&B) is the preferred technique when solving to optimality the HFS problem. Most research so far, however, has concentrated on simplified versions of the problem. The simplest scenario, for example, considers only two stages with a single machine at the first stage and two identical machines in the second stage ($m = 2$, $M^{(1)} = 1$, $M^{(2)} = 2$). For this specific case, the earliest known B&B algorithm was proposed by [155]. Much later, [21] studied the same problem and approached it with B&B, heuristics, and genetic algorithms. Another exact method for this problem, but without waiting allowed between the two stages is given in [61]. The opposite case ($m = 2$, $M^{(1)} = 2$, $M^{(2)} = 1$) was studied by [14] and also by [128]. Problems with two stages and any number of identical parallel machines at the second stage have been recently studied as well. [112] proposed a B&B method with the minimization of total tardiness. Problem instances of up to 15 jobs were shown to be solvable in reasonable times. The case where stage one may have any number of machines and stage two only one is studied in [63]. The authors proposed a B&B that is able to obtain good solutions in a reasonable time. In [70], the 2-stage regular HFS (unconstrained number of machines in stages 1 and 2) with makespan criterion is solved with a very effective B&B method that produces optimal solutions for problems up to 1000 jobs in size. However, the proposed algorithm could not solve many medium instances (20-50 jobs) and in some cases the observed average gap reached more than 4%. [43] are the first to approach two and 3-stage HFSs with uniform parallel machines at each stage with B&B methods. [161] studied the no-wait HFS problem variant. The proposed B&B explores only permutation sequences and jobs are assigned to the earliest available machine at each stage. The author employed dynamic programming for instances of a small size. Recently, [40] have studied a 2-stage problem with multiple identical parallel machines at each stage for the minimization of tardy jobs. The authors propose a B&B method as well as some ad-hoc heuristics.

The earliest known B&B method for the general HFS problem, with any number of stages and any number of parallel machines per stage, is due to [26]. The tree structure that they proposed is an adaptation of that first presented in [29] for the single stage parallel machines problem, and has been the most widely used when dealing with an indefinite number of stages. Despite proposing sophisticated lower bounds, at the time of [26], instances of a very limited size could be solved to optimality. More specifically, problems with up to eight jobs and two stages with three parallel machines each could be solved within several hours of CPU time. [154] also studied the same problem and although it did not compare results directly with those of [26], the performance of the later algorithm seems better. In a similar paper, [153] work with the same methodology but in this case with the minimization of the flowtime. An interesting aspect from [154] and [153] is that only the sequencing problem is solved via B&B and jobs are assigned to the first available machine in each stage. An m-stage HFS with some production planning decisions was considered in [58]. The authors proposed a B&B method that was able to solve only some small instances to optimality. The original lower bound in [26] was further improved in [206] by better exploiting the available information of the partial schedule at a given node. More advanced lower bounds to that in [26] have also been presented in [163] or more recently, in [202].

In most B&B algorithms the construction of a solution starts in stage 1, then it moves to stage 2 and so on until stage m. In [31], a different strategy was adopted. At every decision point the critical stage, i.e., the bottleneck, and a job are selected. The problem is then stated as to decide if there is a schedule with a makespan value smaller than a certain upper bound, UB. In [137], this procedure was enhanced with the so called "energetic reasoning", in which a set of satisfiability tests are performed in order to decide whether there is a schedule that can be constructed at a given node with $C_{\max} \leq D = (LB + UB)/2$. If there is such a schedule, D is calculated with the new UB, otherwise, D is set to $D + 1$. In both cases the search moves to a new node. This algorithm solved instances of up to 15 jobs and 10 stages more efficiently than the rest of available methods. Another successful strategy is to incorporate heuristic non-exact methods into the functioning of B&B algorithms. For example, in [150] and [129] heuristics were used to generate upper bound values. Dispatching rules were used to generate the initial UB and genetic algorithm at the beginning of each stage. In [130], heuristic lower and upper bounds are also extensively used and, in [15] heuristic rules were used for initializing the root node for the flowtime criterion.

Some authors, including [177], have implicitly used B&B through mathematical programming, i.e., they represent their problem as an MIP model and use a regular solver to obtain a solution. [53] also employed mathematical programming to model an HFS with a particular production cost minimization objective. It has to be noted that the literature on chemical engineering has been neglected in the scheduling literature, although it includes some notable papers. A good example is [120], where many mathematical models are given for the m-stage HFS with no-wait and/or limited storage, batching, identical as well as unrelated parallel machines and several optimization criteria. [169] modeled a flexible flow line with blocking and reentry. This model was later improved by [170]. Related papers by the same author appear in [171] and [173]. [145] proposed a mathematical model and heuristics for a complex circuit-packaging problem with sequence dependent setup times. [172] proposes yet more mathematical models for a flexible flow line for tardiness related criteria. A regular HFS, this time with the sum of weighted completion times criterion, is approached with mathematical programming and lagrangian relaxation in [189]. The same problem and objective, but with the addition of limited buffers, is studied in [188]. A rescheduling problem that considers inventory constraints is dealt with in [166]. Some mathematical models are proposed. The HFS is modeled as a resource constrained multi-project scheduling problem with setup times in [209]. Apart from a mathematical formulation, some heuristics are displayed.

Despite the relative success of exact algorithms, they are still incapable of solving medium and large instances and are too complex for real world problems. It is necessary to study non-exact but efficient heuristics. The reader is referred to [204] and [96] for more detailed descriptions of B&B algorithms.

4.2 Heuristics

The simplest type of heuristics are dispatching rules, also known as scheduling policies or list scheduling algorithms. These are simple rules of thumb for the ranking and assignment of jobs onto machines. A number of papers have been exclusively dedicated to their study and comparison on a variety of problems. For instance [25] compares 10 dispatching rules for the m-stage problem with the maximum tardiness criterion. Later, in [28], simulation studies are carried out to further analyze the performance of dispatching rules for the same problem with the makespan and the maximum tardiness objectives. Simulation was also employed to analyze the effectiveness of dispatching rules in [57]. Another comparison study of dispatching rules both in static as well as in dynamic HFSs is given in [92]. Further study of dynamic HFSs and dispatching rules has been given recently in [111]. [125] used data envelopment analysis from the results of [92] as a means to obtain accurate information about the performance of dispatching rules.

Dispatching rules have been used as solution tools for many problems of theoretical and practical relevance. In [59], for instance, dispatching rules and tailored heuristics were used to minimize makespan and maximum tardiness on the m-stage problem. For the same problem, but with the possibility of stage skipping, [97] and [98] propose alternative dispatching rules and a local search algorithm, respectively. The online version of the same problem, with tardiness criterion, and no buffers, was briefly studied in [186] with dispatching rules. [66] and [182] proposed a set of dispatching rules based heuristics for the 2-stage problem investigated in [62]. [198] also studied a 2-stage HFS with a single batching machine at the first stage and machine eligibility in the second stage. [81] and [82] studied the constrained HFS where there is a limit on the maximum number of concurrent jobs. They evaluated a series of dispatching rules under different due date criteria, makespan and flowtime. The m-stage problem with the weighted tardiness objective was approached by [106] using dispatching rules. [203] study an m-stage problem with uniform parallel machines and identical jobs. They investigate the performance of dispatching rules, some tailored heuristics, and derive lower bounds. [84] explore the idea of using different dispatching rules at different stages of the shop.

Dispatching rules are particularly suitable to deal with complex, dynamic, and unpredictable environments and hence their popularity in practice. As early as in [144] a simplified 2-stage HFS arising in the glass container industry was studied and the author proposed several ad-hoc dispatching rules. In [134], the authors applied dispatching rules to a simple 2-stage problem from cable manufacturing with only one machine in the first stage. The authors tested problems with up to 70 jobs and up to four machines in the second stage. Extensions to these dispatching rules were later presented in [133] and [93]. Real problems are also studied in [197] and [2] where a photographic film and a paper bag factories, respectively, were investigated. In both cases, scheduling systems based on dispatching rules were designed. A real rubber production problem was studied by [227]. Another real problem, this time based on textile manufacturing, was approached by [143] with several ad-hoc heuristics. A SMT circuit board flexible flow line was analyzed in [148]. Further results in this line were shown in [149]. The authors consider family setup times and study several dispatching rules. [5] studied a complex HFS problem arising in car manufacturing where different performance criteria were studied.

A number of sophisticated heuristics use a divide-and-conquer strategy in which the original problem is divided into smaller sub-problems that are solved one at a time and their solutions are integrated into a whole solution to the original problem. In [201], for example, a 2-stage problem is simplified into a series of multiple flowshops in order to reduce routing flexibility and to reduce manufacturing costs. [185] studies another 2-stage problem and divides it into two single parallel machines problems, one per stage, where the release times in the second stage are the completion times of the jobs at the first stage. Particularly effective divide-and-conquer strategies for the m-stage problem are the variants of the Shifting Bottleneck Procedure (SBP). These work under the principle of giving full priority to the bottleneck stage, maximizing in this way its productivity, and consequently the productivity of the entire shop. In [39] an SBP approach is proposed for the m-stage problem with the makespan criterion. [226] proposes an SBP heuristic for the minimization of the total weighted tardiness, also for an m-stage problem. While most of bottleneck exploiting methods divide the problem into stages, i.e., they schedule one stage at a time, in [146] this is done by jobs. Every time that a job has to be scheduled, the bottleneck is recalculated, a job selected, and all its operations scheduled. This heuristic, named the progressive bottleneck improvement, competed well with the SBP variant presented in [39]. This algorithm, however, was only tested on HFS problems with 2 and 3 stages. Other authors exploiting the SBP idea are [1] for the makespan criterion, [113] for total tardiness objective and [34] for the number of tardy jobs. Recently [35], the same authors have proposed similar heuristics but for the total tardiness objective. In [162], an interesting heuristic called Flowmult was presented. Flowmult explores $n!$ permutations of the job indices, each representing an order in which operations are prioritized at the first stage of the shop. The rest of the shop is constructed using a first in first out policy, assigning jobs at each stage onto the machine that allows them the fastest completion time. Flowmult was tested on a large set of small instances and found the optimum makespan on 90% of the occasions. On 98% of the cases it found solutions within 5% of the optimum. Many metaheuristics have been designed based on the Flowmult idea, they will be discussed in the next section. The idea of separating the sequencing and machine assignment has been subsequently exploited in many studies. For example, [60] worked with the 2-stage HFS and proposed heuristics that in a first step define a job ordering and then assign jobs at every stage following this ordering. The Flowmult method was again revisited in [164] where specific regular flowshop heuristics were applied to obtain initial job sequences, and in [192], where the m-stage HFS problem with blocking is studied. [27] also separated both decision problems, sequencing and machine assignment, to minimize makespan and flowtime in an m-stage problem.

The vast majority of work in non-exact approaches are tailored heuristics to specific cases of the problem, mainly with 2 and 3 stages. [62], for instance, proposed a very simple heuristic for an HFS with only 2-stages, parallel machines in the first stage and only one machine at the second stage. The same problem was studied later by [63] who also proposed heuristic methods and a B&B algorithm. A similar problem was studied later with the total number of tardy jobs criterion in [68]. In [205] some mathematical programming formulations and heuristics are provided for a simple two stage HFS with only one machine at the second stage. [95] have recently proposed heuristics for a similar 2-stage problem with release dates and a product mix ratio constraint. A similar problem with lot streaming and the total flowtime criterion was studied by [233] and later by [119] for the makespan criterion. [196] tackled the issue of missing operations at the first stage and proposed some simple heuristics. Another simplified setting is that studied in [140], where a problem with two stages with identical parallel machines and unit processing times is approached with heuristic methods. [157] study the case where there are only two uniform machines at the second stage. [67] studied the problem with a single machine in the first stage, any number of identical parallel machines in the second stage, and separable and sequence independent setup and removal times with the makespan criterion. [33] proposed heuristics and provided worst-case analyses for the same problem and [200] study the case when machines at the second stage are unrelated. The 2-stage case with any number of uniform parallel machines is studied in [181] and later revisited by [108]. [30] considers a problem with a set of initial batches, and jobs have a predefined sequence inside each batch. The authors proposed a number of approaches based on heuristics originally designed to tackle the TSP problem. A second article that investigates batching is [19] where two stages with the second one containing batching machines is solved using tailored heuristics. Batching is also studied in [199], where a 2-stage cyclic HFS with product-family batches is considered. [115] also considers batch production allowing split on the single machine at stage 1 and proposed several heuristic methods. The same problem with two stages and identical machines in the first stage was studied by [109]. The authors proposed a fast heuristic running in $\mathcal{O}(n \log n)$, derived error bounds and proposed some lower bounds. [73] consider a 2-stage HFS with the makespan criterion where there is a single machine in the first stage and multiple assembly machines at the second stage. Mathematical models and heuristics are proposed. Similarly, [184] tackle a 2-stage assembly shop with two independent machines in the first stage and two assembly identical machines in the second stage. Dispatching rules, heuristics, and B&B methods are proposed. Assembly stages are also considered in [110] which studied problems with one machine either at the first or at the second stage of the shop and proposed a number of heuristic procedures. A mathematical formulation for the same 2-stage assembly problem and makespan was given in [74] along with some heuristic methods. A two stages problem with unrelated parallel machines in the first stage is considered in [124]. [48] considered a variation of the same problem where preemption of jobs is allowed, and some limited resources (besides machines) are needed at the first stage. Recently, [49], the same author has proposed more heuristics, together with metaheuristics including simulated annealing and genetic algorithms for the same problem. A problem which considers unrelated parallel machines and proportionate processing times is studied by [183]. Similarly, [43] study two and 3-stage problems with uniform parallel machines. They propose lower bounds, polynomial algorithms and metaheuristics. [79] study the uniform parallel machines case, but only at the second stage. They propose several heuristics. Two and 3-stage problems are approached by [99] with the aid of simple methods with known worst-case bounds. The symmetric problem with unavailability constraints was approached in [12] with different heuristics and a B&B algorithm. The no-wait constraint was considered in a 2-stage problem and approached with tailored heuristics by [222].

Some highly peculiar 2-stage problem variants approached heuristically are for example, the transfer batch 2-stage HFS scheduling problem with both separable and inseparable setup times studied in [94]. [101] study the problem where jobs can either be fully processed in one stage (single machine problems) or at both stages and provide some heuristic methods. Another interesting paper is that of [174] where a minimum makespan 2-stage HFS with identical parallel machines is studied. However, in this case, the number of machines at each stage is also part of the problem data. Some heuristics and approximation schemes are provided. Other special HFSs are multiprocessor problems where operations require to be processed by more than one machine simultaneously. [141] proposed heuristics for a 2-stage multiprocessor HFS. In order to deal with uncertainty, in [76] a 2-stage problem with makespan criterion is modeled using fuzzy processing times and approached using fuzzy heuristics. A similar problem but with unavailability periods for the machines is approached by [221].

A relatively small number of heuristics have been proposed for variants of the m-stage problem. A number of such heuristics, for instance, are proposed in [44] and [167] for the m-stage problem with any number of machines per stage. Later, [165] proposed an improvement heuristic for the m-stage HFS with the makespan criterion. [167] proposes a simple heuristic for the minimization of makespan in a similar shop but with limited buffers. Heuristics for the HFS with m-stages, uniform parallel machines and makespan criterion are due to [176] and [107]. [223] study the case where the last of m stages is a batching stage. The results over a large set of small instances indicated that the proposed heuristic obtains close to optimal solutions for very small problem instances. An m-stage with unrelated parallel machines problem is studied in [179]. On-line methods are given in [72] for the m-stage HFS with the makespan criterion. Group scheduling for an m-stage HFS was approached by [121]. [64] consider a realistic problem where due date assignment is part of the problem. The authors also consider limited production resources besides the machines. [22] considered precedence relationships between jobs and proposed several heuristics. [23] studied a fairly complex problem with lags, setup and removal times and precedence constraints with the maximum lateness criterion. An m-stage problem, with job recirculation, common job due dates, and the weighted tardiness criterion was analyzed and solved using a number of heuristics, including dispatching rules, in [41]. [77] extends the work in [76] on fuzzy heuristics for HFS with uncertain data by extending the application of the proposed fuzzy heuristics to the m stages case.

Most real world applications of tailored heuristics are for relatively simple problems with two or three stages only. In [217] a real electronics production system is studied. The author considers three stages, identical parallel machines per stage, and the feature that jobs can skip one or two of the processing stages. The author proposes a pseudo-dynamic programming heuristic. To the best of our knowledge, [217] is the first study in the HFS literature where more than one objective is considered. More precisely, the author minimizes $C_{\max}$ and among the minimum $C_{\max}$ sequences, work in progress (WIP) is considered (lexicographical optimization). In [218], blocking is added to the same application of [217]. Another interesting application of heuristics appeared in [42], where a 2-stage petrochemical production problem with a limited buffer in between the stages was studied. In [46], civil engineering projects are modeled as three stage HFSs with preemption and precedence constraints among jobs. The optimization criterion considered is total tardiness and some simple heuristic methods are proposed. Another reality-inspired problem is approached by [156]. The problem has three stages with one or two machines per stage. The authors presented both exact as well as heuristic algorithms. In a related and later work, the same problem, this time with any number of identical parallel machines per stage, is studied by [180]. Later, [100] studied the same problem and proposed improved heuristics with best known performances. [116] studied a real label sticker manufacturing problem modeled as a 2-stage HFS with sequence dependent setup times and machine eligibility. [160] have studied a complex problem from the ceramic tile sector with unrelated parallel machines, sequence dependent setup times, machine eligibility, machine release dates, positive and negative lags, stage skipping, non-anticipatory setups and job precedence constraints. Some mathematical models and heuristics are proposed. Group technology for setup family grouping in the ceramic tile sector was studied by [13] for the case of three stages.

4.3 Metaheuristics

For the last 20 years, researchers in the combinatorial optimization community have developed successful generic strategies to improve on the performance of simple deterministic heuristics. Most of these methods, known as metaheuristics, add an element of randomness into deterministic heuristics with the idea that its repeated usage may lead to better solutions than the ones generated deterministically. In Simulated Annealing (SA), for example, a local search heuristic is called many times. Randomness is integrated into the acceptance criterion of non-improving solutions with the idea of preventing the stagnation on a local optima, which is the weakness of common steepest descent local search procedures. Tabu Search (TS) [52] achieves the same purpose by allowing non-improving movements and avoiding certain improving movements that may take the search to an already visited point. Genetic algorithms (GA) [126], maintain a "population" of solutions and carry out a simultaneous exploration of different parts of the search space. These three metaheuristics are, by far, the most widely used in HFS scheduling, and probably in scheduling in general. Other metaheuristics that have also been used in HFS are Ant Colony Optimization (ACO), Artificial Immune Systems (AIS), Neural Networks (NN) and others.

The key of successful metaheuristics lays on the reduction of the explored space. A sophisticated approach, particularly successful on the makespan objective, exploits the simple observation that in order to improve a schedule, one or more jobs that are part of the critical path of the schedule have to be reassigned. This means that any modification in a schedule that does not involve a job in the critical path is at best not worse than the current schedule. The space explored can be further pruned by taking into account many other properties of the graphical representation of the problem. These are thoroughly discussed in [136] and later in [135]. In [136], these properties were used in a TS approach which showed very good results for the makespan objective. In [226] an intricate metaheuristic involving the Shifting Bottleneck Procedure (SBP) and a randomized local search procedure were used to minimize the weighted tardiness objective.

Most metaheuristics, however, reduce the explored space using a far simpler strategy: to restrict the search to the space of job permutations. The idea is to find a permutation of the n jobs and build a schedule by assigning jobs onto the machines according to this ordering. One of the earliest papers to exploit this idea is [208], where a dispatching rule was used to seed a TS algorithm to solve a 2-stage HFS with a single machine in the second stage and stage-based sequence dependent setup times. [71] also studied the 2-stage case but with identical parallel machines at each stage. The authors presented a lower bound, a TS heuristic, and an SA algorithm. Both methods work over the permutation of jobs and a simple heuristic is used for job assignment at each stage. [36] presented a TS heuristic for a two stage flexible flow line with makespan criterion. Another TS was proposed by [55] in a no-wait HFS inspired by a real manufacturing problem. In [216], the capacity between stages is limited (*block*). A TS that explores the permutation space was developed and compared with the approaches proposed in [218] and in [167]. TS outperformed both heuristics. In [38] and [37] a similar TS was also developed and used to solve real instances of a container handling system. The multiprocessor task problem with precedence relationships is approached by [142]. A TS metaheuristic is proposed, some special cases of the problem are proved to be polynomially solvable, whereas others are proved to be $\mathcal{NP}$-hard. A simplification of the HFS in which machine assignments are known in advance, was also approached with TS by [50]. The authors minimize the sum of earliness and tardiness. An m-stage problem with group scheduling is approached in [122] with TS. Yet another application of TS is for the regular HFS with the sum of the weighted completion times criterion and limited buffers, [215]. The same objective, but with all jobs having the same processing times on all machines, is solved in [80] using a column generation strategy and constructive heuristics. No much later, the same authors improved on their results with a constructive GA [178]. There are few studies that consider unrelated parallel machines at each stage. An example is [123], that proposed a SA method for this problem with total flowtime minimization. He also considered sequence dependent removal times (R_{sd}) and sequence independent setup times. [8] presented several heuristics and a SA method for an m-stage HFS that models client-server requests. Another SA method with the same solution representation was given in [88] together with a lower bound to evaluate algorithm performance. This bound was proved wrong and corrected later by [69]. Simulated annealing has been used to solve a HFS with sequence dependent setup times and transportation times between stages of an automated guided vehicle in [131]. In a similar paper [132], SA is applied to the same problem but without transportation constraints.

An interesting randomized local search heuristic is due to [114]. In there, randomness is introduced by modifying the problem data. Each variant of the original problem is solved with a deterministic heuristic and the solution is evaluated using the original data. The idea is, of course, that the solution to one of the modified problems improves on that generated with the data of the original problem. The authors concluded that the method is easy to adapt to cope with different objective functions. In the same paper, the authors proposed a lower bound for the makespan criterion which was later improved by [103]. Other authors have used a representation where a permutation for each machine at each stage is maintained. This is sometimes referred to as operation processing order or exact representation. [83] employ this scheme and propose several TS and SA methods. In order to cope with such large search space, the schedule is constructed, at each stage, after the solution of a linear program. Similarly, in [54] a SA algorithm that does not separate the job sequencing and machine assignment decisions is applied to an m-stage unrelated parallel machine problem.

Genetic algorithms have also been widely used. In [220], GA was used to search the permutation space for the solution of the m-stage problem with makespan as objective. The same problem, with the addition of sequence dependent setup times was approached in [105] using a GA with the Random Keys representation proposed in [18]. The proposed RKGA outperformed several other specialized heuristics including those of [104]. In [139], a multi-processor problem was considered. A GA, similar to the one presented in [220], was compared with the sophisticated TS of [142] and obtained better results. A realistic problem from a check-processing company that considered recirculation and the sum of weighted tardiness objective was approached using GA by [20]. In [159], a GA was employed to minimize makespan on an m-stage problem with unrelated parallel machines, sequence dependent setup times and machine eligibility. The proposed GA was superior to a wide range of heuristics and other metaheuristics, among them, ACO based heuristics, TS procedures, other GAs, SA and deterministic procedures. The proposed GA also obtained better schedules than the ones generated manually by the personnel of a real world ceramic tiles production shop. A similar GA was recently proposed for the same problem but with the additional consideration of limited buffers by [228]. A similar problem, with unrelated parallel machines at each stage, and setup times, was approached in [89] with GAs and later in [90] with several heuristics including dispatching rules, tailored heuristics, GA, TS and SA. In these two papers, the authors study a linear combination of the makespan and the number of tardy jobs as an objective. Another real world application is given in [87], where a real printed circuit board manufacturing system modeled as a 3-stage HFS is approached with GA. The HFS with multiprocessor tasks is studied for the makespan criterion by [175] and by [139] and approached with GA. A GA and a SA were proposed for a fairly complex cyclic scheduling flexible flow line with lot sizing in [85]. Another complex problem with production time windows in flexible flow lines is exposed in [151]. In [210], the authors use GA to explore the permutation space for the first stage of the shop and dispatching rules others than the first in first out for the rest. They carried out experiments on different objective functions and obtained good results, among them, improved on the performance of the SBP with local search presented in [226] for the sum of weighted tardiness objective. Later, the same authors use a permutation for the first stage of the shop and combinations of dispatching rules for the rest. Both the permutation and the combination of heuristics are found by GA. A similar GA is shown in [211], where several criteria are jointly considered. A simplistic GA approach was proposed in [219] for an HFS with machine eligibility constraints and makespan criterion. The same SMT circuit board flexible flow line studied in [148, 149] has been studied more recently in [91], where a mathematical programming based solution approach, dispatching rules and a TS algorithm are proposed. Very recently, a memetic algorithm (GA with embedded local search) was proposed for a similar problem [191].

Other metaheuristics have been used less frequently. Artificial Immune Systems (AIS), for instance, have been used in [47] and in [232]. In the later case, the proposed AIS outperformed the GA proposed in [105]. Neural Networks were used in [65] as a mechanism to decide among several heuristics, for the one to be used on a particular instance. Another application of NN is given in [214] where it was used for the m-stage HFS with the makespan criterion. However, the proposed approach is complex and the results are relatively poor. A similar work with NN is due to [187]. In both works, unfortunately, comparisons are carried out only against simple heuristics. The addition of sequence dependent setup times is shown in [190]. Their results, together with those in [214] and [187] suggest that NN, at least as proposed by the authors, are not as successful as the other reviewed metaheuristics. [230] proposed an Ant Colony Optimization (ACO) metaheuristic for the multiprocessor task problem with precedence relationships and showed superior results than [142]. A Particle Swarm Optimization (PSO) metaheuristic, which allegedly improved upon the results in [230], was proposed by [195]. The same problem is approached with tailored heuristics in [231] and with an iterated greedy metaheuristic in [229]. [6] studied the regular m-stage HFS with makespan criterion. They proposed an ACO method which produced better results than a curtailed B&B algorithm. A radically different approach, namely an agent system method, was used in [16] for a three stage identical parallel machine HFS.

Simulation tools, which are adequate to model the inherent complexity of real world problems, have been used in combination with common metaheuristics. For example, TS has been combined with simulation in [224] to solve a real multi-layer ceramic capacitor production problem. The same problem was approached using GA and simulation tools in [225]. A similar problem is approached by [102] also with the aid of simulation tools. Recently, [7] also mix simulation tools with TS to solve a very complex scheduling problem. [158] consider an m-stage HFS with the makespan criterion. The authors use simulation tools for the planning and batching stages and SA for the scheduling phase. Another realistic problem approach with SA and simulation tools is due to [11].

5 Analysis of the literature

This review has examined more than 200 papers, mainly dealing with the HFS problem and its many variants. As with other fields of study, the number of papers being published has been steadily raising over the past few decades, as Figure 1 shows.

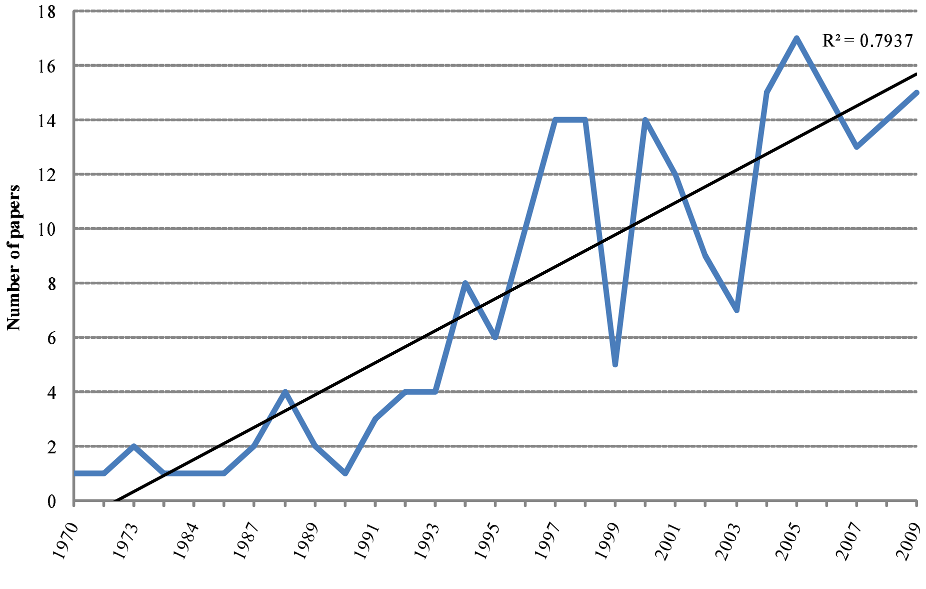

Figure 1: Evolution of number of papers per year

As can be seen, there is a clear increasing trend which shows the growing interest in this field. It is reasonable to expect that in the coming years the HFS problem will receive an even larger amount of attention.

There are, however, some important remarks to be made. Table 2 shows the percentage of papers that deal with 2, 3 or m-stage problems and whether the machines at each stage are identical, uniform or unrelated.

Table 2: Percentage of the reviewed papers according to number of stages and type of parallel machines

Number of	Type of parallel machines			
stages	Identical	Uniform	Unrelated	Total
2	25.12	1.86	4.65	31.63
3	4.19	1.4	0	5.59
m	54.41	1.4	6.97	62.78
Total	83.72	4.66	11.62	100.00

As shown, a fourth part of the reviewed literature deals with simple 2-stage problems with identical parallel machines and almost a third only tackles 2-stage problems. While these problems are of theoretical interest, many times, the developed methods are not easily extendible to three or more stages. Similarly, a large percentage of the reviewed papers consider identical machines at each stage (83.72%) and only a meager 6.97% of the literature tackles m-stage problems with unrelated parallel machines at each stage. It is clear that the m-stage problem with unrelated parallel machines is the most general case and therefore, the most likely to be found in practice. As a matter of fact, from the reviewed literature, most papers dealing with real problems do so with m stages and unrelated parallel machines.

Similarly, in Figure 2 we separate the reviewed literature among the different objective functions. Notice that "Other" includes cost functions and/or problem or situation specific objective functions.

Clearly, the literature is heavily biased towards the $C_{\max}$ criterion with a 60% of the references studying this single objective. Total/average completion time or flowtime, both in their unweighted and weighted forms, add up another 11%. It is striking to see that from all surveyed papers, only a total of 1% deal with the earliness-tardiness criterion, which is so important for real problems. Another relevant observation is that only a handful of papers deal with multiple objectives, and, to the best of our knowledge, the papers dealing with more than one objective do so separately. We are only aware of the recent papers of [89] and [90] that consider a weighted sum of objectives. Multi-objective scheduling is a very rich field of study as recent works show [193]. For regular flowshop problems, the number of existing multi-objective approaches is large as reviewed by [127]. Therefore, multi-objective scheduling for HFS is a necessary venue of research that has

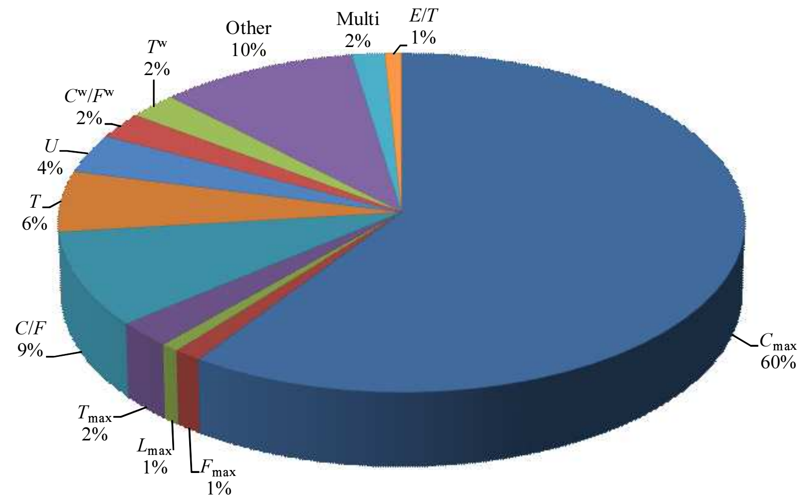

Figure 2: Distribution of objective functions
C/F=Total/average completion or flow time, T=Total/average tardiness, E/T=Total/Average sum of earliness and tardiness, Multi=Multiples objectives

not been explored so far.

It is also interesting to study the different methodologies and techniques that the authors apply in the reviewed literature. Figure 3 shows a pie chart with this distribution.

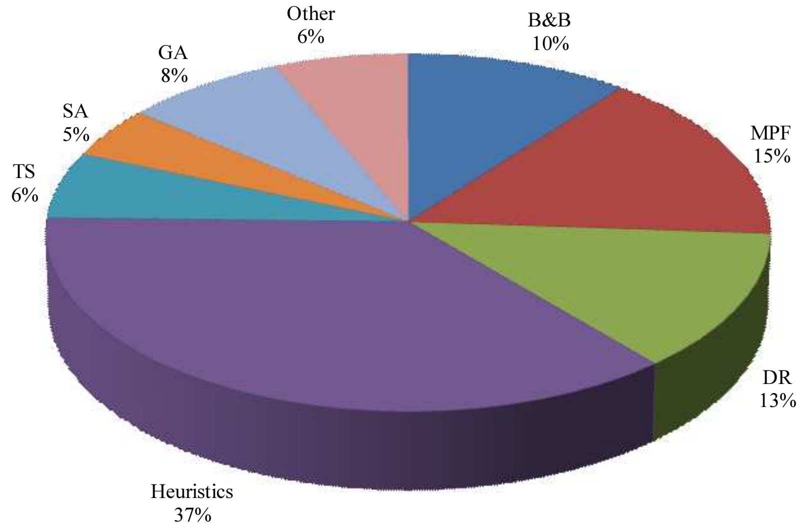

Figure 3: Distribution of employed techniques
B&B=Branch and Bound, MPF=Mathematical Programming and formulation, DR=Dispatching Rules, TS=Tabu Search, SA=Simulated Annealing, GA=Genetic Algorithms

First of all, adding up B&B and mathematical programs and models we have a full 25% of reviewed papers. We have to consider that these techniques have proven so far useful either for simplified problems, specific settings and/or small problems so a larger focus is needed on approaches able to solve more general and larger problems. Under heuristics we have classified many different algorithms and ad-hoc methods that are specific and do not contain a well known metaheuristic template. Most times these methods are single-pass heuristics able to solve large problems but with a limited performance. Together with dispatching rules, these methods add up a 50% of the reviewed papers. This means that only the remaining 25% is left for SA, GA, TS and other metaheuristic techniques like ACO, PSO and other recent methodologies. Clearly, there is a large opportunity for research here. Metaheuristics have long ago established themselves as state-of-the-art methodologies for the vast majority of scheduling problems and therefore, HFS should be no exception.

6 Research opportunities and conclusions

In this paper we have reviewed and analyzed more than 200 papers dealing with the hybrid flowshop (HFS) or related variants. This field of study is attracting more research efforts due to the many applications that this realistic problem setting has in practice. In the review, we have classified all the papers according to many parameters, including problem variant studied, constraints, objective functions and employed methodologies. We are certain that this review work will be helpful for other researchers in the area as well as for establishing a reference starting point for new research efforts.

In practice, objectives vary and hence a variety of HFS models are possible. It is unrealistic for the minimization of $C_{\max}$ to match all cases. Nevertheless, 60% of the reviewed papers are exclusively concerned with it. A very small percentage of the remaining papers, on the other hand, are dedicated to the solution of problems with real world motivated functions. This imbalance seems to be unjustified. Minimizing makespan may be relevant in several cases since it optimizes the use of limited resources. However, there are other objectives that in practice are sensible too. For instance, minimizing holding costs (inventory costs) may be more relevant than minimizing makespan, or to meet the clients demands on time, or both of them at the same time. Unfortunately, it is not feasible to study all possible cost functions that could arise in practice. The same situation occurs with the constraints and assumptions, it is unlikely that a real world problem exactly matches any of the models intensively studied in the literature. It seems to be a more promising strategy to generate heuristics which show flexibility on a wide range of HFS problems.

It is also important to consider that the real world is unpredictable and dynamic. Algorithms must be able to find solutions which remain robust under different scenarios. No results have been found on robust scheduling in HFS. Moreover, the equally important problem of rescheduling has not received the attention it deserves. In both cases, technologies that have been developed to address such problems in other scheduling scenarios, such as job shop [86], should be adapted to HFS.

Production scheduling problems are Multi-Objective (MO) by nature, which means that several criteria, in conflict with each other, have to be considered at a time. Research in MO optimization is concerned with the generation of solutions in which none of the objective functions can be improved without paying a cost in other objective(s) (usually referred to as non-dominated solutions). In [218], makespan and the inventory in process were considered simultaneously, however, the heuristic developed concentrated on minimizing makespan and the inventory in process was treated as a secondary objective. No attempt on finding non-dominated solutions in HFS has been reported in the literature to the best of our knowledge.

Setup times in scheduling have recently attracted a lot of attention. The recent review paper of [9] and the study of setup importance from [10] are just some examples. Apart from some isolated reviewed papers, sequence dependent setup times have been scarcely studied in HFS settings and more research is needed in this regard.

Finally, more effective metaheuristic templates are being proposed. The bulk of this research, however, only concentrates on relatively simple combinatorial optimization problems. A concentrated effort is required in order to apply these recent methodologies to complex HFS problems.

Acknowledgements

Rubén Ruiz is partially funded by the Spanish Ministry of Science and Innovation, under the projects "SMPA - Advanced Parallel Multiobjective Sequencing: Practical and Theorerical Advances" with reference DPI2008-03511/DPI and "OACS - Advanced Optimization of the Supply Chain" with reference IAP-020100-2008-11.

Appendix

Table 3: Summary of reviewed papers

Year	Ref.	Problem	Comments
1970	[155]	$FH2,((1^{(1)},P2^{(2)}))\vert\vert C_{\max}$	B&B, small instances
1971	[14]	$FH2,((P2^{(1)},1^{(2)}))\vert\vert C_{\max}$	B&B, small instances
1973	[161]	$FHm,((PM^{(k)})_{k=1}^{m})\vert no-wait\vert C_{\max}$	B&B, small instances
	[128]	$FH2,((P2^{(1)},1^{(2)}))\vert\vert C_{\max}$	B&B, small instances
1979	[144]	$FH2,((PM^{(k)})_{k=1}^{2})\vert no-idle^{(1)}\vert\{\bar{T},\bar{U}\}$	dispatching rules

Table 3: Summary of reviewed papers

Year	Ref.	Problem	Comments
1984	[134]	$FH2,((1^{(1)},R2^{(2)}))\vert\vert\text{waiting and idleness}$	dispatching rules
1985	[217]	$FH3,((PM^{(k)})_{k=1}^{3})\vert skip\vert C_{\max}$	MPF and heuristics, flow lines
1987	[133]	$FH2,((1^{(1)},R2^{(2)}))\vert\vert\text{waiting and idleness}$	dispatching rules
	[97]	$FHm,((PM^{(k)})_{k=1}^{m})\vert block,brkdwn,S_{nsd},skip\vert\bar{F}$	dispatching rules, flow lines
1988	[218]	$FH3,((PM^{(k)})_{k=1}^{3})\vert block,skip\vert\{C_{\max},WIP\}$	pseudo DP, circuit board manufacturing, flow lines
	[24]	$FHm,((PM^{(k)})_{k=1}^{m})\vert\vert C_{\max}$	B&B, MPF
	[62]	$FH2,((P2^{(1)},1^{(2)}))\vert\vert C_{\max}$	heuristics, NP-hard proof
	[98]	$FHm,((PM^{(k)})_{k=1}^{m})\vert block,brkdwn,S_{nsd}\vert\bar{F}$	dispatching rules, local search
1989	[182]	$FH2,((PM^{(k)})_{k=1}^{2})\vert\vert C_{\max}$	dispatching rules based heuristics
	[227]	$FH3,((P2^{(1)},1^{(2)},R5^{(3)}))\vert skip,revisit\vert C_{\max}$	dispatching rules
1990	[177]	$FH2,((P10^{(1)},P12^{(2)}))\vert\vert\text{ad-hoc}$	MPF, paper industry
1991	[26]	$FHm,((PM^{(k)})_{k=1}^{m})\vert\vert C_{\max}$	B&B
	[66]	$FH2,((1^{(1)},PM^{(2)}))\vert\vert C_{\max}$	dispatching rules based heuristics
	[143]	$FHm,((PM^{(k)})_{k=1}^{m})\vert S_{sd}\vert\{C_{\max},\bar{T}\}$	ad-hoc heuristics, textile industry
1992	[81]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert\{\bar{T},\bar{U}\}$	restricted jobs. dispatching rules
	[153]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert prmu\vert\bar{F}$	B&B
		$FH2,((PM^{(k)})_{k=1}^{2}))\vert\vert\bar{F}$	heuristics
	[154]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert prmu\vert C_{\max}$	B&B
1993	[2]	$FHm,((RM^{(k)})_{k=1}^{m})\vert S_{sd}\vert\bar{T}^{w}$	dispatching rules, paper bags factory
	[167]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert buffer,skip\vert C_{\max}$	heuristics, flow lines
	[197]	$FH2,((PM^{(k)})_{k=1}^{2}))\vert\vert\text{ad-hoc}$	dispatching rules, photographic film production
	[208]	$FH2,((PM^{(1)},1^{(2)}))\vert S_{sd}\vert C_{\max}$	TS and heuristics
1994	[42]	$FH2,((PM^{(1)},PM^{(2)}))\vert buffer\vert C_{\max}$	heuristics, petrochemical production
	[32]	$FHm,((PM^{(k)})_{k=1}^{m})\vert block\vert\bar{E}^{w}+\bar{T}^{w}$	MPF based heuristic
	[82]	$FHm,((PM^{(k)})_{k=1}^{m})\vert\vert\{C_{\max},\bar{F},F_{\max}\}$	restricted jobs. dispatching rules
	[44]	$FHm,((PM^{(k)})_{k=1}^{m})\vert skip\vert C_{\max}$	heuristics, flow lines
	[53]	$FHm,((PM^{(k)})_{k=1}^{m})\vert\vert\text{Cost}$	MPF, specific problem
	[67]	$FH2,((1^{(1)},PM^{(2)}))\vert S_{nsd},R_{nsd}\vert C_{\max}$	heuristics
	[109]	$FH2,((PM^{(k)})_{k=1}^{2})\vert\vert C_{\max}$	heuristics, bounds
	[148]	$FHm,((PM^{(k)})_{k=1}^{m})\vert\vert\bar{F}$	dispatching rules
1995	[163]	$FHm,((PM^{(k)})_{k=1}^{m})\vert\vert C_{\max}$	lower bounds
	[162]	$FHm,((PM^{(k)})_{k=1}^{m})\vert\vert C_{\max}$	MPR heuristic
	[33]	$FH2,((PM^{(1)},1^{(2)}))\vert\vert C_{\max}$	heuristics, worst-case performance
		$FH2,((1^{(1)},PM^{(2)}))\vert\vert C_{\max}$	heuristics, worst-case performance
	[3]	$FHm,((RM^{(k)})_{k=1}^{m})\vert S_{sd}\vert\{C_{\max},F_{\max},\bar{F}\}$	MPF, heuristics, carpet manufacturing
	[168]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert blocking,skip\vert C_{\max}$	heuristics, flow lines
	[200]	$FH2,((1^{(1)},RM^{(2)}))\vert\vert\{C_{\max},F_{\max}\}$	heuristics
1996	[25]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert T_{\max}$	dispatching rules
	[46]	$FH3,((PM^{(k)})_{k=1}^{m}))\vert pmtn,prec\vert\bar{T}$	heuristics, civil engineering
	[60]	$FH2,((PM^{(k)})_{k=1}^{m}))\vert\vert C_{\max}$	heuristics, lower bounds
	[59]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert C_{\max},T_{\max}$	heuristics, dispatching rules
	[75]	$FHm,((PM^{(k)})_{k=1}^{m})\vert no-idle\vert C_{\max}$	MPF, heuristics
	[78]	$FH2,((PM^{(k)})_{k=1}^{2}))\vert pmtn\vert C_{\max}$	NP-hard proof
	[164]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert C_{\max}$	heuristics
	[198]	$FH2,((1^{(1)},RM^{(2)}))\vert M_{j}^{(2)}\vert C_{\max},WIP,CU$	dispatching rules, lot streaming
	[205]	$FH2,((PM^{(1)},1^{(2)}))\vert pmtn^{(1)}\vert T_{\max}$	MPF, heuristics
	[207]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert\bar{C}$	B&B
1997	[5]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert r_{j},other\vert\text{several}$	real car production problem, dispatching rules
	[22]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert prec\vert C_{\max}$	heuristics
	[58]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert skip\vert C_{\max}$	production planning, MPF, B&B
	[63]	$FH2,((PM^{(1)},1^{(2)}))\vert\vert C_{\max}$	B&B, heuristics
	[71]	$FH2,((PM^{(k)})_{k=1}^{2}))\vert\vert C_{\max}$	lower bound, SA, TS
	[93]	$FH2,((1^{(1)},R2^{(2)}))\vert\vert\text{waiting and idleness}$	dispatching rules. Note for [133]
	[92]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert\text{several}$	dispatching rules
	[94]	$FH2,((PM^{(k)})_{k=1}^{2}))\vert S_{nsd}\vert C_{\max}$	transfer batch problem, heuristics
	[114]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert skip\vert\{C_{\max},\bar{T}^{w}\}$	problem space based local search
	[115]	$FH2,((1^{(1)},PM^{(2)}))\vert batch,S_{sd},split\vert C_{\max}$	heuristics
	[140]	$FH2,((PM^{(k)})_{k=1}^{2}))\vert size_{jk},p_{j}=1\vert C_{\max}$	heuristics, lower bounds
	[183]	$FH2,((RM^{(k)})_{k=1}^{2}))\vert proportional\vert C_{\max}$	MPF, heuristics
	[185]	$FH2,((RM^{(k)})_{k=1}^{2}))\vert\vert C_{\max}$	heuristics
	[206]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert C_{\max}$	B&B, new lower bound

Table 3: Summary of reviewed papers

Year	Ref.	Problem	Comments
1998	[28]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert\{C_{\max},\bar{T}\}$	dispatching rules
	[36]	$FH2,((PM^{(k)})_{k=1}^{2})\vert skip\vert C_{\max}$	TS
	[43]	$FH(m\leq 3),((QM^{(k)})_{k=1}^{m}))\vert\vert C_{\max}$	polynomial algorithm when $m=2$, B&B and heuristics when $m=3$
	[68]	$FH2,((1^{(1)},PM^{(2)}))\vert\vert\bar{U}$	heuristics
	[72]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert C_{\max}$	heuristics
	[79]	$FH2,((1^{(1)},Q2^{(2)}))\vert\vert C_{\max}$	heuristics
	[101]	$FH2,((PM^{(k)})_{k=1}^{2}))\vert\vert C_{\max}$	complex routings, heuristics
	[110]	$FH2,((P1^{(1)},1^{(2)}))\vert assembly^{(2)}\vert C_{\max}$	heuristics, also symmetric problem
	[136]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert C_{\max}$	TS
	[149]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert\bar{F}$	dispatching rules
	[150]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert C_{\max}$	hybrid B&B-GA
	[156]	$FH3,(1^{(1)},R2^{(2)},1^{(3)})\vert M_j^{(2)}\vert C_{\max}$	B&B and heuristics
	[226]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert\bar{T}^w$	B&B, SBP, hybrid SBP-LS
1999	[125]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert several$	DEA analysis over [92]
	[27]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert\{C_{\max},\bar{F}\}$	flow shop based heuristics
	[54]	$FHm,((RM^{(k)})_{k=1}^{m}))\vert\vert C_{\max}$	SA
	[57]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert\{C_{\max},\bar{F},WIP,CU\}$	simulation model, dispatching rules
	[117]	$FH***\vert*\vert*$	review paper
	[203]	$FHm,((QM^{(k)})_{k=1}^{m}))\vert r_j\vert C_{\max}$	dispatching rules, heuristics, NP-hard proof
	[204]	$FH***\vert*\vert*$	review paper
2000	[23]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert prec,lags,S_{nsd},R_{nsd}\vert L_{\max}$	heuristics
	[31]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert C_{\max}$	B&B
	[55]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert no-wait\vert C_{\max}$	TS, concrete blocks production
	[65]	$FH2,((1^{(1)},PM^{(2)}))\vert\vert C_{\max}$	Neural networks for choosing heuristics
	[76]	$FH2,((PM^{(k)})_{k=1}^{2}))\vert\vert C_{\max}$	fuzzy heuristics, fuzzy processing times
	[84]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert\text{several}$	dispatching rules
	[99]	$FH\{m=2,3\},((PM^{(k)})_{k=1}^{m}))\vert\vert C_{\max}$	heuristics
	[118]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert S_{sd},block\vert\bar{E}^w+\bar{T}^w$	MPF based heuristics
	[130]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert C_{\max}$	B&B
	[169]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert skip,block,reentry\vert C_{\max}$	MPF, flow lines
	[174]	$FH2,((PM^{(k)})_{k=1}^{2}))\vert\vert C_{\max}$	heuristics, machine number part of input
	[199]	$FH2,((1^{(1)},PM^{(2)}))\vert\vert C_{\max}$	heuristics
	[201]	$FH2,((PM^{(k)})_{k=1}^{2}))\vert\vert C_{\max}$	heuristics, problem simplification
	[220]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert C_{\max}$	MPR-GA
2001	[15]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert barT$	B&B
	[39]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert\{C_{\max},L_{\max}\}$	SB heuristic
	[74]	$FH2,((P1^{(1)},1^{(2)}))\vert assembly^{(2)}\vert C_{\max}$	heuristics, also MPF
	[77]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert C_{\max}$	fuzzy heuristics, fuzzy processing times
	[103]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert skip\vert C_{\max}$	lower bound
	[106]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert\bar{T}^w$	dispatching rules
	[135]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert C_{\max}$	LS, SA, TS
	[137]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert C_{\max}$	B&B
	[158]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert C_{\max}$	SA, planning and simulation
	[165]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert C_{\max}$	heuristics
	[170]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert skip,block,reentry\vert C_{\max}$	MPF, flow lines
	[180]	$FH3,((PM^{(k)})_{k=1}^{3}))\vert\vert C_{\max}$	heuristics
2002	[4]	$FH2,((PM^{(k)})_{k=1}^{2}))\vert\vert\text{non-regular}$	MPF, heuristics, photographic film production
	[45]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert pmtn,prec\vert C_{\max}$	polynomial algorithm for special case of $prec$ ([78])
	[64]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert\bar{E}^u+\bar{T}^v+\bar{C}^w+\bar{d}^z$	heuristic, assignable due dates, controllable processing times
	[73]	$FH2,((1^{(1)},Pm^{(2)}))\vert assembly^{(2)}\vert C_{\max}$	MPF, heuristics
	[87]	$FH3,((PM^{(k)})_{k=1}^{3}))\vert\vert C_{\max}$	GA, printed circuit boards
	[157]	$FH2,((1^{(1)},Q2^{(2)}))\vert\vert C_{\max}$	MPF, DP, heuristics, NP-complete proof
	[171]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert skip,block,reentry\vert C_{\max}$	MPF, flow lines
	[173]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert skip,block,reentry\vert C_{\max}$	MPF, flow lines
	[176]	$FHm,((QM^{(k)})_{k=1}^{m}))\vert\vert C_{\max}$	heuristics
2003	[104]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert S_{sd}\vert C_{\max}$	heuristics
	[116]	$FH2,((RM^{(k)})_{k=1}^{2}))\vert S_{sd}^{(1)},M_j^{(2)}\vert wT_{\max}$	heuristic, label sticker manufacturing
	[141]	$FH2,((PM^{(k)})_{k=1}^{m}))\vert size_{jk}\vert wT_{\max}$	heuristics, lower bounds
	[146]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert C_{\max}$	bottleneck exploiting heuristic

Table 3: Summary of reviewed papers

Year	Ref.	Problem	Comments
	[181]	$FH2,((QM^{(k)})^2_{k=1}))\vert\vert C_{\max}$	heuristics, lower bounds
	[214]	$FHm,((PM^{(k)})^m_{k=1}))\vert\vert C_{\max}$	Neural Networks
	[219]	$FHm,((RM^{(k)})^m_{k=1}))\vert M_j\vert C_{\max}$	GA
2004	[1]	$FHm,((PM^{(k)})^m_{k=1}))\vert\vert C_{\max}$	bottleneck exploiting heuristic
	[11]	$FHm,((PM^{(k)})^m_{k=1}))\vert avail\vert several$	simulation, heuristics, SA
	[16]	$FH3,((PM^{(k)})^3_{k=1}))\vert\vert C_{\max}$	agent-based approach
	[20]	$FHm,((PM^{(k)})^m_{k=1}))\vert recrc\vert\bar{U}^w$	MPF, GA, lower bounds,checks processing
	[47]	$FHm,((PM^{(k)})^m_{k=1}))\vert\vert C_{\max}$	Artificial Immune Systems
	[91]	$FHm,((PM^{(k)})^m_{k=1}))\vert blocking, skip\vert C_{\max}$	flow lines, MPF, TS, huristics
	[105]	$FHm,((PM^{(k)})^m_{k=1}))\vert S_{sd}\vert C_{\max}$	MPF, MPR-GA
	[112]	$FH2,((1^{(1)},PM^{(2)}))\vert\vert\bar{T}$	B&B
	[113]	$FHm,((PM^{(k)})^m_{k=1}))\vert\vert\bar{T}$	bottleneck exploiting heuristic
	[142]	$FHm,((PM^{(k)})^m_{k=1}))\vert size_{jk},prec\vert C_{\max}$	TS, polynomial and NP-hard cases proofs
	[175]	$FHm,((PM^{(k)})^m_{k=1}))\vert size_{jk}\vert C_{\max}$	MPR-GA
	[192]	$FHm,((PM^{(k)})^m_{k=1}))\vert no-wait\vert C_{\max}$	MPR heuristics
	[216]	$FHm,((PM^{(k)})^m_{k=1}))\vert block\vert C_{\max}$	MPR-TS
	[222]	$FH2,((PM^{(k)})^2_{k=1}))\vert no-wait\vert C_{\max}$	heuristics
	[224]	$FHm,((PM^{(k)})^m_{k=1}))\vert\vert C_{\max}$	TS-simulation, ceramic capacitor manufacturing
2005	[13]	$FH3,((PM^{(k)})^3_{k=1})\vert S_{sd}\vert other$	MPF, group technology
	[21]	$FH2,((1^{(1)},P2^{(2)}))\vert\vert C_{\max}$	B&B, GA, heuristics
	[41]	$FHm,((PM^{(k)})^m_{k=1}))\vert recrc\vert\bar{T}^w$	dispatching rules, heuristics
	[61]	$FH2,((1^{(1)},P2^{(2)}))\vert no-wait,(p_j=1)^1\vert C_{\max}$	exact method
	[96]	$FHm,((PM^{(k)})^m_{k=1}))\vert\vert\{C_{\max},\bar{C}\}$	review on exact solution methods
	[121]	$FHm,((PM^{(k)})^m_{k=1}))\vert\vert C_{\max}$	group scheduling, setups, heuristics
	[123]	$FHm,((RM^{(k)})^m_{k=1}))\vert S_{nsd},R_{sd}\vert\bar{C}$	SA
	[129]	$FHm,((PM^{(k)})^m_{k=1}))\vert\vert C_{\max}$	hybrid B&B-GA
	[139]	$FHm,((PM^{(k)})^m_{k=1}))\vert size_{jk}\vert C_{\max}$	MPR-GA
	[145]	$FHm,((PM^{(k)})^m_{k=1}))\vert S_{sd}\vert\bar{T}$	MPF, heuristics
	[172]	$FHm,((PM^{(k)})^m_{k=1}))\vert skip\vert\{barT,T_{\max}\}$	MPF, flow lines
	[186]	$FHm,((PM^{(k)})^m_{k=1})\vert skip\vert\bar{T}$	dispatching rules, flow lines
	[187]	$FHm,((PM^{(k)})^m_{k=1}))\vert\vert C_{\max}$	Neural Networks
	[190]	$FHm,((PM^{(k)})^m_{k=1}))\vert S_{sd}\vert C_{\max}$	Neural Networks
	[202]	$FHm,((PM^{(k)})^m_{k=1})\vert\vert C_{\max}$	lower bounds
	[210]	$FHm,((PM^{(k)})^m_{k=1})\vert r_j\vert\{C_{\max},T_{\max},\bar{T}^w,\bar{C}\}$	several variants of MPR-GA
	[213]	$FH***\vert skip,*\vert *$	review paper
	[221]	$FH2,((PM^{(k)})^2_{k=1})\vert avail\vert C_{\max}$	approximation algorithms
	[233]	$FH2,((PM^{(1)},1^{(2)}))\vert\vert\bar{F}$	heuristics, bounds, lot streaming
2006	[8]	$FHm,((PM^{(k)})^m_{k=1}))\vert\vert\bar{C}$	MPR-SA
	[12]	$FH2,((1^{(1)},PM^{(2)}))\vert avail\vert C_{\max}$	B&B, heuristics, complexity
	[38]	$FH3,((RM^{(k)})^3_{k=1}))\vert prec,block,S_{nsd}\vert C_{\max}$	MPR-TS
	[70]	$FH2,((PM^{(k)})^2_{k=1})\vert\vert C_{\max}$	B&B
	[88]	$FHm,((PM^{(k)})^m_{k=1}))\vert\vert C_{\max}$	MPR-SA, lower bounds
	[107]	$FHm,((QM^{(k)})^m_{k=1}))\vert\vert C_{\max}$	FS based heuristics
	[108]	$FH2,((QM^{(k)})^2_{k=1}))\vert\vert C_{\max}$	heuristics, lower bounds. Note for [181]
	[122]	$FHm,((PM^{(k)})^m_{k=1}))\vert\vert C_{\max}$	group scheduling, TS
	[159]	$FHm,((RM^{(k)})^m_{k=1}))\vert S_{sd},M_j\vert C_{\max}$	MPR-GA
	[179]	$FHm,((RM^{(k)})^m_{k=1}))\vert buffer\vert several$	heuristics
	[189]	$FHm,((PM^{(k)})^m_{k=1})\vert\vert\bar{F}^w$	MPF, lagrangian relaxation
	[188]	$FHm,((PM^{(k)})^m_{k=1})\vert buffer\vert\bar{F}^w$	MPF, lagrangian relaxation
	[211]	$FHm,((PM^{(k)})^m_{k=1}))\vert\vert$composite functions	hybrid GA + dispatching rules
	[230]	$FHm,((PM^{(k)})^m_{k=1}))\vert size_{jk}\vert C_{\max}$	ACO
	[232]	$FHm,((PM^{(k)})^m_{k=1}))\vert S_{sd}\vert C_{\max}$	Artificial Immune System
2007	[6]	$FHm,((PM^{(k)})^m_{k=1}))\vert\vert C_{\max}$	Ant Colony Optimization
	[30]	$FH2,((P2^{(1)},1^{(2)}))\vert batch\vert C_{\max}$	TSP-based heuristics
	[37]	$FH3,((RM^{(k)})^3_{k=1}))\vert S_{sd},block,prec\vert C_{\max}$	MPF, lower bounds, TS
	[50]	$FHm,((PM^{(k)})^m_{k=1}))\vert assign\vert\bar{E}T$	TS, special problem
	[83]	$FHm,((PM^{(k)})^m_{k=1}))\vert r_j\vert Cost$	TS, SA, heuristics
	[85]	$FHm,((RM^{(k)})^m_{k=1}))\vert lot,skip\vert Cost$	GA, SA, flow lines
	[100]	$FH3,((PM^{(k)})^3_{k=1}))\vert\vert C_{\max}$	heuristics
	[151]	$FHm,((PM^{(k)})^m_{k=1}))\vert batch,skip\vert\{\bar{F},Cost\}$	GA, SA, flow lines
	[151]	$FH***\vert *\vert *$	review paper

Table 3: Summary of reviewed papers

Year	Ref.	Problem	Comments
	[166]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert r_j, skip\vert\bar{U}$	MPF, inventory constraints, rescheduling
	[212]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert r_j, d_j\vert$several	FS based heuristics
	[209]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert S_{sd}\vert\bar{T}^w$	MPF, DR, heuristics. Multi-project RCPSP
	[225]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert$several	GA, simulation
	[223]	$FHm,((PM^{(1)},PM^{(2)}))\vert batch^{(m)}\vert\bar{F}$	MPF, lagrangian relaxation, heuristics
2008	[34]	$FHm,((RM^{(k)})_{k=1}^{m}))\vert skip\vert\bar{U}$	heuristics, flow lines
	[48]	$FH2,((RM^{(1)},1^{(2)}))\vert pmtn, resources^{(1)}\vert C_{\max}$	heuristics
	[69]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert C_{\max}$	lower bounds, note over [88]
	[80]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert p_j\vert\bar{F}^w$	proportionate shop. heuristics, column generation
	[89]	$FHm,((RM^{(k)})_{k=1}^{m}))\vert S_{sd}, r_j\vert\alpha C_{\max}+(1-\alpha)\bar{U}$	MPF, heuristics, GA, SA, TS
	[102]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert\vert$several	heuristics, simulation
	[119]	$FH2,((PM^{(1)},1^{(2)}))\vert\vert C_{\max}$	heuristics, bounds, lot streaming
	[120]	$FHm,((RM^{(k)})_{k=1}^{m}))\vert no-wait, S_{nsd}\vert several$	MPF
	[124]	$FH2,((RM^{(1)},1^{(2)}))\vert M_j\vert C_{\max}$	heuristics, eligibility on first stage
	[160]	$FHm,((RM^{(k)})_{k=1}^{m}))\vert skip, rm, lag, S_{sd}, M_j, prec\vert C_{\max}$	MPF, heuristics
	[178]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert p_j\vert\bar{F}^w$	proportionate shop. GA
	[184]	$FH2,((2^{(1)},P2^{(2)}))\vert assembly^{(2)}\vert\bar{F}$	heuristics
	[195]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert size_{jk}\vert C_{\max}$	Particle Swarm Optimization
	[196]	$FH2,((1^{(1)},P2^{(2)}))\vert skip^{(1)}\vert C_{\max}$	heuristics
2009	[90]	$FHm,((RM^{(k)})_{k=1}^{m}))\vert S_{sd}, r_j\vert\alpha C_{\max}+(1-\alpha)\bar{U}$	MPF, heuristics, dispatching rules, GA
	[95]	$FH2,((PM^{(1)},1^{(2)}))\vert\vert C_{\max}$	heuristics, product-mix
	[191]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert skip, block, reentry\vert C_{\max}$	GA mixed with LS
	[229]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert size_{jk}\vert C_{\max}$	Iterated Greedy (IG)
	[19]	$FH2,((PM^{(1)},PM^{(2)}))\vert batch^{(2)}\vert C_{\max}$	heuristics
	[215]	$FHm,((PM^{(k)})_{k=1}^{m})\vert buffer\vert\bar{F}^w$	TS
	[111]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert r_j\vert\bar{F}$	dispatching rules
	[131]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert S_{sd}, transport\vert\{\bar{F},\bar{T}\}$	SA
	[228]	$FHm,((RM^{(k)})_{k=1}^{m}))\vert S_{sd}, M_j, buffer\vert C_{\max}$	MPR-GA
	[35]	$FHm,((RM^{(k)})_{k=1}^{m}))\vert skip\vert\bar{T}$	heuristics, flow lines
	[49]	$FH2,((RM^{(1)},1^{(2)}))\vert pmtn, resources^{(1)}\vert C_{\max}$	heuristics, GA, SA
	[40]	$FH2,((PM^{(k)})_{k=1}^{m}))\vert\vert\bar{U}$	B&B, heuristics
	[132]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert S_{sd}\vert\{\bar{F},\bar{T}\}$	SA
	[7]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert S_{sd}, reentry, batch\vert several$	simulation, heuristics, TS
	[231]	$FHm,((PM^{(k)})_{k=1}^{m}))\vert size_{jk}\vert C_{\max}$	heuristics

Mathematical Programming Formulation (MPF), Dynamic Programming (DP), Tabu Search (TS)
Branch and Bound (B&B), Multiple Permutation Representation(MPR), Work In Progress (WIP)
Local Search (LS), Shifting Bottleneck Procedure (SBP), Simulated Annealing (SA)

References

[1] M. J. Acero-Domínguez and C. D. Patermina-Arboleda. Scheduling jobs on a k-stage flexible flow shop using a toc-based (bottleneck) procedure. In M. H. Jones, S. D. Patek, and B. E. Tawney, editors, *Proceedings of the 2004 Systems and Information Engineering Design Symposium*, pages 295–298. IEEE press, 2004.

[2] L. Adler, N. Fraiman, E. Kobacker, M. Pinedo, J. C. Plotnicoff, and T. P. Wu. Bpss: A scheduling support system for the packaging industry. *Operations Research*, 41(4):641–648, 1993.

[3] E.-H. Aghezzaf, A. Artiba, O. Moursli, and C. Tahon. Hybrid flowshop problems, a decomposition based heuristic approach. In *Proceedings of the International Conference on Industrial Engineering and Production Management, IEPM'95*, pages 43–56. FUCAM-INRIA, 1995.

[4] E.-H. Aghezzaf and H. Van Landeghem. An integrated model for inventory and production planning in a two-stage hybrid production system. *International Journal of Production Research*, 40(17):4323–4339, 2002.

[5] A. Agnetis, A. Pacifici, F. Rossi, M. Lucertini, S. Nicoletti, F. Nicolo, G. Oriolo, D. Pacciarelli, and E. Pesaro. Scheduling of flexible flow lines in an automobile assembly plant. *European Journal of Operational Research*, 97(2):348–362, 1997.

[6] K. Alaykyran, O. Engin, and A. Doyen. Using ant colony optimization to solve hybrid flow shop

scheduling problems. *International Journal of Advanced Manufacturing Technology*, 35(5-6):541–550, 2007.

[7] A. Alfieri. Workload simulation and optimisation in multi-criteria hybrid flowshop scheduling: a case study. *In press at International Journal of Production Research*, 2009.

[8] A. Allahverdi and F. S. Al-Anzi. Scheduling multi-stage parallel-processor services to minimize average response time. *Journal of the Operational Research Society*, 57(1):101–110, 2006.

[9] A. Allahverdi, C. T. Ng, T. C. E. Cheng, and M. Y. Kovalyov. A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187(3):985–1032, 2008.

[10] H. M. Allahverdi, A.and Soroush. The significance of reducing setup times/setup costs. *European Journal of Operational Research*, 187(3):978–984, 2008.

[11] H. Allaoui and A. Artiba. Integrating simulation and optimization to schedule a hybrid flow shop with maintenance constraints. *Computers & Industrial Engineering*, 47(4):431–450, 2004.

[12] H. Allaoui and A. Artiba. Scheduling two-stage hybrid flow shop with availability constraints. *Computers & Operations Research*, 33(5):1399–1419, 2006.

[13] C. Andres, J. M. Albarracin, G. Tormo, E. Vicens, and J. P. Garcia-Sabater. Group technology in a hybrid flowshop environment: A case study. *European Journal of Operational Research*, 167(1):272–281, 2005.

[14] T. S. Arthanary and K. G. Ramaswamy. An extension of two machine sequencing problem. *OPSEARCH: The Journal of the Operational Research Society of India*, 8(4):10–22, 1971.

[15] M. Azizoglu, E. Cakmak, and S. Kondakci. A flexible flowshop problem with total flow time minimization. *European Journal of Operational Research*, 132(3):528–538, 2001.

[16] A. Babayan and D. He. Solving the n-job 3-stage flexible flowshop scheduling problem using an agent-based approach. *International Journal of Production Research*, 42(4):777–799, 2004.

[17] O. O. Balogun and K. Popplewell. Towards the integration of flexible manufacturing system scheduling. *International Journal of Production Research*, 37(15):3399–3428, 1999.

[18] J. C. Bean. Genetic algorithms and random keys for sequencing and optimization. *Informs Journal on Computing*, 6(2):154–160, 1994.

[19] A. Bellanger and A. Oulamara. Scheduling hybrid flowshop term with parallel batching machines and compatibilities. *Computers & Operations Research*, 36(6):1982–1992, 2009.

[20] S. Bertel and J.-C. Billaut. A genetic algorithm for an industrial multiprocessor flow shop scheduling problem with recirculation. *European Journal of Operational Research*, 159(3):651–662, 2004.

[21] A. Bolat, I. Al-Harkan, and B. Al-Harbi. Flow-shop scheduling for three serial stations with the last two duplicate. *Computers & Operations Research*, 32(3):647–667, 2005.

[22] V. Botta-Genoulaz. Considering bills of material in hybrid flow shop scheduling problems. In *1997 IEEE International Symposium on Assembly and Task Planning, ISATP 97*, pages 194–199. IEEE Press, 1997.

[23] V. Botta-Genoulaz. Hybrid flow shop scheduling with precedence constraints and time lags to minimize maximum lateness. *International Journal of Production Economics*, 64(1-3):101–111, 2000.

[24] S. A. Brah. *Scheduling in a flow shop with multiple processors*. Thesis/dissertation, University of Houston, 1988.

[25] S. A. Brah. A comparative analysis of due date based job sequencing rules in a flow shop with multiple processors. *Production Planning & Control*, 7(4):362–373, 1996.

[26] S. A. Brah and J. L. Hunsucker. Branch and bound algorithm for the flow-shop with multiple processors. *European Journal of Operational Research*, 51(1):88–99, 1991.

[27] S. A. Brah and L. L. Loo. Heuristics for scheduling in a flow shop with multiple processors. *European Journal of Operational Research*, 113(1):113–122, 1999.

[28] S. A. Brah and G. E. Wheeler. Comparison of scheduling rules in a flow shop with multiple processors: A simulation. *Simulation*, 71(5):302–311, 1998.

[29] P. Bratley, M. Florian, and P. Robillard. Scheduling with earliest start and due date constraints on multiple machines. *Naval Research Logistics Quearterly*, 22(1):165–173, 1975.

[30] P. Caricato, A. Grieco, and D. Serino. Tsp-based scheduling in a batch-wise hybrid flow-shop. *Robotics and Computer-Integrated Manufacturing*, 23(2):234–241, 2007.

[31] J. Carlier and E. Néron. An exact method for solving the multi-processor flow-shop. *RAIRO*

Recherche Opérationnelle, 34(1):1–25, 2000.

[32] S.-C. Chang and D.-Y. Liao. Scheduling flexible flow shops with no setup effects. *IEEE Transactions on Robotics and Automation*, 10(2):112–122, 1994.

[33] B. Chen. Analysis of classes of heuristics for scheduling a two-stage flow-shop with parallel machines at one stage. *Journal of the Operational Research Society*, 46(2):234–244, 1995.

[34] C. L. Chen and C. L. Chen. Bottleneck-based heuristics to minimize tardy jobs in a flexible flow line with unrelated parallel machines. *International Journal of Production Research*, 46(22):6415–6430, 2008.

[35] C. L. Chen and C. L. Chen. Bottleneck-based heuristics to minimize total tardiness for the flexible flow line with unrelated parallel machines. *In press at Computers & Industrial Engineering*, 2009.

[36] C. L. Chen, J. M. Usher, and N. Palanimuthu. A tabu search based heuristic for a flexible flow line with minimum flow time criterion. *International Journal of Industrial Engineering-Theory Applications and Practice*, 5(2):157–168, 1998.

[37] L. Chen, N. Bostel, P. Dejax, J. C. Cai, and L. F. Xi. A tabu search algorithm for the integrated scheduling problem of container handling systems in a maritime terminal. *European Journal of Operational Research*, 181(1):40–58, 2007.

[38] L. Chen, L.-F. Xi, J.-G. Cai, N. Bostel, and P. Dejax. An integrated approach for modeling and solving the scheduling problem of container handling systems. *Journal of Zhejiang University SCIENCE A*, 7(2):234–239, 2006.

[39] J. L. Cheng, Y. Karuno, and H. Kise. A shifting bottleneck approach for a parallel-machine flowshop scheduling problem. *Journal of the Operations Research Society of Japan*, 44(2):140–156, 2001.

[40] H. S. Choi and D. H. Lee. Scheduling algorithms to minimize the number of tardy jobs in two-stage hybrid flow shops. *In press at Computers & Industrial Engineering*, 2009.

[41] S.-W. Choi, Y.-D. Kim, and G.-C. Lee. Minimizing total tardiness of orders with reentrant lots in a hybrid flowshop. *International Journal of Production Research*, 43(11):2149–2167, 2005.

[42] D. E. Deal, T. Yang, and S. Hallquist. Job scheduling in petrochemical production: 2-stage processing with finite intermediate storage. *Computers & Chemical Engineering*, 18(4):333–344, 1994.

[43] M. M. Dessouky, M. I. Dessouky, and S. K. Verma. Flowshop scheduling with identical jobs and uniform parallel machines. *European Journal of Operational Research*, 109(3):620–631, 1998.

[44] F. Y. Ding and D. Kittichartphayak. Heuristics for scheduling flexible flow lines. *Computers & Industrial Engineering*, 26(1):27–34, 1994.

[45] H. Djellab and K. Djellab. Preemptive hybrid flowshop scheduling problem of interval orders. *European Journal of Operational Research*, 137(1):37–49, 2002.

[46] M. Dror and P. A. Mullaserif. Three stage generalized flowshop: Scheduling civil engineering projects. *Journal of Global Optimization*, 9(3-4):321–344, 1996.

[47] O. Engin and A. Doyen. A new approach to solve hybrid flow shop scheduling problems by artificial immune system. *Future Generation Computer Systems*, 20(6):1083–1095, 2004.

[48] E. Figielska. A new heuristic for scheduling the two-stage flowshop with additional resources. *Computers & Industrial Engineering*, 54(4):750–763, 2008.

[49] E. Figielska. A genetic algorithm and a simulated annealing algorithm combined with column generation technique for solving the problem of scheduling in the hybrid flowshop with additional resources. *In press at Computers & Industrial Engineering*, 2009.

[50] D. A. Finke, D. J. Medeiros, and M. T. Traband. Multiple machine jit scheduling: a tabu search approach. *International Journal of Production Research*, 45(21):4899–4915, 2007.

[51] M. R. Garey and D. S. Johnson. *Computers and intractability: A guide to the theory of NP-Completeness.* A Series of books in the mathematical sciences. W.H. Freeman, San Francisco, 1979.

[52] F. Glover and M. Laguna. *Tabu search.* Kluwer Academic Publishers, Boston, 1997.

[53] W. B. Gooding, J. F. Pekny, and P. S. Mccroskey. Enumerative approaches to parallel flowshop scheduling via problem transformation. *Computers & Chemical Engineering*, 18(10):909–927, 1994.

[54] M. Gourgand, N. Grangeon, and S. Norre. Metaheuristics for the deterministic hybrid flow shop problem. In *Proceedings of the International Conference on Industrial Engineering and Production Management, IEPM'99*, pages 136–145. FUCAM-INRIA, 1999.

[55] J. Grabowski and J. Pempera. Sequencing of jobs in some production system. *European Journal of Operational Research*, 125(3):535–550, 2000.

[56] R. L. Graham, E. L. Lawler, J.K. Lenstra, and A. H. G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, 5:287–326, 1979.

[57] N. Grangeon, A. Tanguy, and N. Tchernev. Generic simulation model for hybrid flow-shop. *Computers & Industrial Engineering*, 37(1-2):207–210, 1999.

[58] K. R. Gue, G. L. Nemhauser, and M. Padron. Production scheduling in almost continuous time. *IIE Transactions*, 29(5):391–398, 1997.

[59] A. G. P. Guinet and M. M. Solomon. Scheduling hybrid flowshops to minimize maximum tardiness or maximum completion time. *International Journal of Production Research*, 34(6):1643–1654, 1996.

[60] A. G. P. Guinet, M. M. Solomon, P. K. Kedia, and A. Dussauchoy. A computational study of heuristics for two-stage flexible flowshops. *International Journal of Production Research*, 34(5):1399–1415, 1996.

[61] S. Guirchoun, P. Martineau, and J.-C. Billaut. Total completion time minimization in a computer system with a server and two parallel processors. *Computers & Operations Research*, 32(3):599–611, 2005.

[62] J. N. D. Gupta. Two-stage, hybrid flow shop scheduling problem. *Journal of the Operational Research Society*, 39(4):359–364, 1988.

[63] J. N. D. Gupta, A. M. A. Hariri, and C. N. Potts. Scheduling a two-stage hybrid flow shop with parallel machines at the first stage. *Annals of Operations Research*, 69:171–191, 1997.

[64] J. N. D. Gupta, K. Kruger, V. Lauff, F. Werner, and Y. N. Sotskov. Heuristics for hybrid flow shops with controllable processing times and assignable due dates. *Computers & Operations Research*, 29(10):1417–1439, 2002.

[65] J. N. D. Gupta, R. S. Sexton, and E. A. Tunc. Selecting scheduling heuristics using neural networks. *Informs Journal on Computing*, 12(2):150–162, 2000.

[66] J. N. D. Gupta and E. A. Tunc. Schedules for a two-stage hybrid flowshop with parallel machines at the second stage. *International Journal of Production Research*, 29(7):1489–1502, 1991.

[67] J. N. D. Gupta and E. A. Tunc. Scheduling a two-stage hybrid flowshop with separable setup and removal times. *European Journal of Operational Research*, 77(3):415–428, 1994.

[68] J. N. D. Gupta and E. A. Tunc. Minimizing tardy jobs in a two-stage hybrid flowshop. *International Journal of Production Research*, 36(9):2397–2417, 1998.

[69] M. Haouari and L. Hidri. On the hybrid flowshop scheduling problem. *International Journal of Production Economics*, 113(1):495–497, 2008.

[70] M. Haouari, L. Hidri, and A. Gharbi. Optimal scheduling of a two-stage hybrid flow shop. *Mathematical Methods of Operations Research*, 64(1):107–124, 2006.

[71] M. Haouari and R. M'Hallah. Heuristic algorithms for the two-stage hybrid flowshop problem. *Operations Research Letters*, 21(1):43–53, 1997.

[72] J. T. Havill and W. Mao. On-line algorithms for hybrid flow shop scheduling problems. In *Proceedings of the Fourth International Conference on Computer Science and Informatics*, pages 134–137, 1998.

[73] D. He and A. Babayan. Scheduling manufacturing systems for delayed product differentiation in agile manufacturing. *International Journal of Production Research*, 40(11):2461–2481, 2002.

[74] D. He, A. Babayan, and A. Kusiak. Scheduling manufacturing systems in an agile environment. *Robotics and Computer Integrated Manufacturing*, 17(1-2):87–97, 2001.

[75] D. W. He, A. Kusiak, and A. Artiba. A scheduling problem in glass manufacturing. *IIE Transactions*, 28(2):129–139, 1996.

[76] T. P. Hong and T. T. Wang. Fuzzy flexible flow shops at two machine centers for continuous fuzzy domains. *Information Sciences*, 129(1-4):227–237, 2000.

[77] T. P. Hong, T. T. Wang, and S. L. Wang. A palmer-based continuous fuzzy flexible flow-shop scheduling algorithm. *International Journal of Soft Computing*, 5(6):426–433, 2001.

[78] J. A. Hoogeveen, J. K. Lenstra, and B. Veltman. Preemptive scheduling in a two-stage multiprocessor flow shop is np-hard. *European Journal of Operational Research*, 89(1):172–175, 1996.

[79] W. Huang and S. Li. A two-stage hybrid flowshop with uniform machines and setup times. *Mathematical and Computer Modelling*, 27(2):27–45, 1998.

[80] Y. M. Huang and D. F. Shiau. Combined column generation and constructive heuristic for a propor-

tionate flexible flow shop scheduling. *International Journal of Advanced Manufacturing Technology*, 38(7-8):691–704, 2008.

[81] J. L. Hunsucker and J. R. Shah. Performance of priority rules in a due date flow-shop. *Omega-International Journal of Management Science*, 20(1):73–89, 1992.

[82] J. L. Hunsucker and J. R. Shah. Comparative performance analysis of priority rules in a constrained flow-shop with multiple processors environment. *European Journal of Operational Research*, 72(1):102–114, 1994.

[83] A. Janiak, E. Kozan, M. Lichtenstein, and C. Oguz. Metaheuristic approaches to the hybrid flow shop scheduling problem with a cost-related criterion. *International Journal of Production Economics*, 105(2):407–424, 2007.

[84] M. S. Jayamohan and C. Rajendran. A comparative analysis of two different approaches to scheduling in flexible flow shops. *Production Planning & Control*, 11(6):572–580, 2000.

[85] M. Jenabi, S. M. T. Fatemi Ghomi, S. A. Torabi, and B. Karimi. Two hybrid meta-heuristics for the finite horizon elsp in flexible flow lines with unrelated parallel machines. *Applied Mathematics and Computation*, 186(1):230–245, 2007.

[86] M. T. Jensen. *Robust and flexible scheduling with evolutionary computation.* PhD thesis, Department of Computer Science, University of Aarhus, 2001.

[87] Z. H. Jin, K. Ohno, T. Ito, and S. E. Elmaghraby. Scheduling hybrid flowshops in printed circuit board assembly lines. *Production and Operations Management*, 11(2):216–230, 2002.

[88] Z. H. Jin, Z. Yang, and T. Ito. Metaheuristic algorithms for the multistage hybrid flowshop scheduling problem. *International Journal of Production Economics*, 100(2):322–334, 2006.

[89] J. Jungwattanakit, M. Reodecha, P. Chaovalitwongse, and F. Werner. Algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. *International Journal of Advanced Manufacturing Technology*, 37(3-4):354–370, 2008.

[90] J. Jungwattanakit, M. Reodecha, P. Chaovalitwongse, and F. Werner. 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, 2009.

[91] W. Kaczmarczyky, T. J. Sawik, A. Schaller, and T. M. Tirpak. Optimal versus heuristic scheduling of surface mount technology lines. *International Journal of Production Research*, 42(10):2083–2110, 2004.

[92] S. N. Kadipasaoglu, W. Xiang, and B. M. Khumawala. A comparison of sequencing rules in static and dynamic hybrid flow systems. *International Journal of Production Research*, 35(5):1359–1384, 1997.

[93] S. N. Kadipasaoglu, W. Xiang, and B. M. Khumawala. A note on scheduling hybrid flow systems. *International Journal of Production Research*, 35(5):1491–1494, 1997.

[94] J. S. Kim, S. H. Kang, and S. M. Lee. Transfer batch scheduling for a two-stage flowshop with identical parallel machines at each stage. *Omega-International Journal of Management Science*, 25(5):547–555, 1997.

[95] Y.-D. Kim, B.-J. Joo, and J.-H. Shin. Heuristics for a two-stage hybrid flowshop scheduling problem with ready times and a product-mix ratio constraint. *In press at Journal of Heuristics*, 2008.

[96] T. Kis and E. Pesch. A review of exact solution methods for the non-preemptive multiprocessor flowshop problem. *European Journal of Operational Research*, 164(3):592–608, 2005.

[97] S. Kochhar and R. J. T. Morris. Heuristic methods for flexible flow line scheduling. *Journal of Manufacturing Systems*, 6(4):299–314, 1987.

[98] S. Kochhar, R. J. T. Morris, and W. S. Wong. The local search approach to flexible flow line scheduling. *Engineering Costs and Production Economics*, 14(1):25–37, 1988.

[99] C. Koulamas and G. J. Kyparisis. Asymptotically optimal linear time algorithms for two-stage and three-stage flexible flow shops. *Naval Research Logistics*, 47(3):259–268, 2000.

[100] C. Koulamas and G. J. Kyparisis. A note on performance guarantees for sequencing three-stage flexible flowshops with identical machines to minimize makespan. *IIE Transactions*, 39(5):559–563, 2007.

[101] P. Kouvelis and G. Vairaktarakis. Flowshops with processing flexibility across production stages. *IIE Transactions*, 30(8):735–746, 1998.

[102] Y. Kuo, T. Yang, C. Cho, and Y. C. Tseng. Using simulation and multi-criteria methods to provide

robust solutions to dispatching problems in a flow shop with multiple processors. *Mathematics and Computers in Simulation*, 78(1):40–56, 2008.

[103] M. E. Kurz and R. G. Askin. An adaptable problem-space-based search method for flexible flow line scheduling. *IIE Transactions*, 33(8):691–693, 2001.

[104] M. E. Kurz and R. G. Askin. Comparing scheduling rules for flexible flow lines. *International Journal of Production Economics*, 85(3):371–388, 2003.

[105] M. E. Kurz and R. G. Askin. Scheduling flexible flow lines with sequence-dependent setup times. *European Journal of Operational Research*, 159(1):66–82, 2004.

[106] G. J. Kyparisis and C. Koulamas. A note on weighted completion time minimization in a flexible flow shop. *Operations Research Letters*, 29(1):5–11, 2001.

[107] G. J. Kyparisis and C. Koulamas. Flexible flow shop scheduling with uniform parallel machines. *European Journal of Operational Research*, 168(3):985–997, 2006.

[108] G. J. Kyparisis and C. Koulamas. A note on makespan minimization in two-stage flexible flow shops with uniform machines. *European Journal of Operational Research*, 175(2):1321–1327, 2006.

[109] C. Y. Lee and Vairaktarakis. G. L. Minimizing makespan in hybrid flowshops. *Operations Research Letters*, 16(3):149–158, 1994.

[110] C. Y. Lee and G. L. Vairaktarakis. Performance comparison of some classes of flexible flow shops and job shops. *International Journal of Flexible Manufacturing Systems*, 10(4):379–405, 1998.

[111] G. C. Lee. Estimating order lead times in hybrid flowshops with different scheduling rules. *In press at Computers & Industrial Engineering*, 2009.

[112] G.-C. Lee and Y.-D. Kim. A branch-and-bound algorithm for a two-stage hybrid flowshop scheduling problem minimizing total tardiness. *International Journal of Production Research*, 42(22):4731–4743, 2004.

[113] Y.-D Lee, G.-C.and Kim and S.-W. Choi. Bottleneck-focused scheduling for a hybrid flowshop. *International Journal of Production Research*, 42(1):165–181, 2004.

[114] V. J. Leon and B. Ramamoorthy. An adaptable problem-space-based search method for flexible flow line scheduling. *IIE Transactions*, 29(2):115–125, 1997.

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

[116] H.-T. Lin and C.-J. Liao. 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, 2003.

[117] R. Linn and W. Zhang. Hybrid flow shop scheduling: A survey. *Computers & Industrial Engineering*, 37(1-2):57–61, 1999.

[118] C.-Y. Liu and S.-C. Chang. Scheduling flexible flow shops with sequence-dependent setup effects. *IEEE Transactions on Robotics and Automation*, 16(4):408–419, 2000.

[119] J. Y. Liu. Single-job lot streaming in m-1 two-stage hybrid flowshops. *European Journal of Operational Research*, 187(3):1171–1183, 2008.

[120] Y. Liu and I. A. Karimi. Scheduling multistage batch plants with parallel units and no interstage storage. *Computers & Chemical Engineering*, 32(4-5):671–693, 2008.

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

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

[123] C. Y. Low. Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines. *Computers & Operations Research*, 32(8):2013–2025, 2005.

[124] C. Y. Low, C. J. Hsu, and C. T. Su. A two-stage hybrid flowshop scheduling problem with a function constraint and unrelated alternative machines. *Computers & Operations Research*, 35(3):845–853, 2008.

[125] Braglia M. and A. Petroni. Data envelopment analysis for dispatching rule selection. *Production Planning & Control*, 10(5):454–461, 1999.

[126] Z. Michalewicz. *Genetic algorithms + data structures = evolution programs*. Springer-Verlag, Berlin, third edition edition, 1996.

[127] G. Minella, R. Ruiz, and M. Ciavotta. A review and evaluation of multi-objective algorithms for the flowshop scheduling problem. *INFORMS Journal on Computing*, 20(3):451–471, 2008.

[128] B. S. Mittal and P. C. Bagga. Two machine sequencing problem with parallel machines. *OPSEARCH : Journal of the Operational Research Society of India*, 10:50–61, 1973.

[129] H. Morita and N. Shio. Hybrid branch and bound method with genetic algorithm for flexible flowshop scheduling problem. *JSME International Journal Series C-Mechanical Systems Machine Elements and Manufacturing*, 48(1):46–52, 2005.

[130] O. Moursli and Y. Pochet. A branch-and-bound algorithm for the hybrid flowshop. *International Journal of Production Economics*, 64(1-3):113–125, 2000.

[131] B. Naderi, M. Zandieh, A. Khaleghi Ghoshe Balagh, and V. Roshanaei. An improved simulated annealing for hybrid flowshops with sequence-dependent setup and transportation times to minimize total completion time and total tardiness. *In press at Expert Systems with Applications*, 2009.

[132] B. Naderi, M. Zandieh, and V. Roshanaei. Scheduling hybrid flowshops with sequence dependent setup times to minimize makespan and maximum tardiness. *In press at International Journal of Advanced Manufacturing Technology*, 2009.

[133] S. L. Narasimhan and P. M. Mangiameli. A comparison of sequencing rules for a two-stage hybrid flowshop. *Decision Sciences*, 18(2):250–265, 1987.

[134] S. L. Narasimhan and S. S. Panwalkar. Scheduling in a two-stage manufacturing process. *International Journal of Production Research*, 22(4):555–564, 1984.

[135] E. G. Negenman. Local search algorithms for the multiprocessor flow shop scheduling problem. *European Journal of Operational Research*, 128(1):147–158, 2001.

[136] E. Nowicki and C. Smutnicki. The flow shop with parallel machines: A tabu search approach. *European Journal of Operational Research*, 106(2-3):226–253, 1998.

[137] E. Néron, P. Baptiste, and J. N. D. Gupta. Solving hybrid flow shop problem using energetic reasoning and global operations. *Omega-International Journal of Management Science*, 29(6):501–511, 2001.

[138] P. J. O'Grady and U. Menon. A concise review of flexible manufacturing systems and FMS literature. *Computers in Industry*, 7(2):155–167, 1986.

[139] C. Oguz and M. Ercan. A genetic algorithm for hybrid flow-shop scheduling with multiprocessor tasks. *Journal of Scheduling*, 8(4):323–351, 2005.

[140] C. Oguz and M. F. Ercan. Scheduling multiprocessor tasks in a two-stage flow-shop environment. *Computers & Industrial Engineering*, 33(1-2):269–272, 1997.

[141] C. Oguz, M. F. Ercan, T. C. E. Cheng, and Y. F. Fung. Heuristic algorithms for multiprocessor task scheduling in a two-stage hybrid flow-shop. *European Journal of Operational Research*, 149(2):390–403, 2003.

[142] C. Oguz, Y. Zinder, V. Do, A. Janiak, and M. Lichtenstein. Hybrid flow-shop scheduling problems with multiprocessor task systems. *European Journal of Operational Research*, 152(1):115–131, 2004.

[143] Guinet. A. G. P. Textile production systems: A succession of non-identical parallel processor shops. *Journal of the Operational Research Society*, 42(8):655–671, 1991.

[144] R. J. Paul. Production scheduling problem in the glass-container industry. *Operations Research*, 27(2):290–302, 1979.

[145] W. L. Pearn, S. H. Chung, M. H. Yang, and C. Y. Chen. The integrated circuit packaging scheduling problem (icpsp): A case study. *International Journal of Industrial Engineering-Theory Applications and Practice*, 12(3):296–307, 2005.

[146] S. Phadnis, J. Brevick, and S. Irani. Development of a new heuristic for scheduling flow-shops with parallel machines by prioritizing bottleneck stages. *Journal of Integrated Design & Process Science*, 7(1):87–97, 2003.

[147] M. Pinedo. *Scheduling: Theory, algorithms, and systems*. Prentice Hall, Upper Saddle, N.J, second edition edition, 2002.

[148] S. Piramuthu, N. Raman, and M. J. Shaw. Learning-based scheduling in a flexible manufacturing flow line. *IEEE Transactions on Engineering Management*, 41(2):172–182, 1994.

[149] S. Piramuthu, N. Raman, and M. J. Shaw. Decision support system for scheduling a flexible flow system: Incorporation of feature construction. *Annals of Operations Research*, 78:219–234, 1998.

[150] M. C. Portmann, A. Vignier, D. Dardilhac, and D. Dezalay. Branch and bound crossed with ga to solve hybrid flowshops. *European Journal of Operational Research*, 107(2):389–400, 1998.

[151] D. Quadt and D. Kuhn. A taxonomy of flexible flow line scheduling procedures. *European Journal*

of Operational Research, 178(3):686–698, 2007.

[152] R. Rachamadugu and K. E. Stecke. Classification and review of FMS scheduling procedures. *Production Planning & Control*, 5(1):2–20, 1994.

[153] C. Rajendran and D. Chaudhuri. A multistage parallel-processor flowshop problem with minimum flowtime. *European Journal of Operational Research*, 57(1):111–122, 1992.

[154] C. Rajendran and D. Chaudhuri. Scheduling in n-jobs, m stage flowshop with parallel processors to minimize makespan. *International Journal of Production Economics*, 27(2):137–143, 1992.

[155] T. B. K. Rao. Sequencing in the order a, b, with multiplicity of machines for a single operation. *OPSEARCH : Journal of the Operational Research Society of India*, 7:135–144, 1970.

[156] F. Riane, A. Artiba, and S. E. Elmaghraby. A hybrid three-stage flowshop problem: Efficient heuristics to minimize makespan. *European Journal of Operational Research*, 109(2):321–329, 1998.

[157] F. Riane, A. Artiba, and S. E. Elmaghraby. Sequencing a hybrid two-stage flowshop with dedicated machines. *International Journal of Production Research*, 40(17):4353–4380, 2002.

[158] F. Riane, A. Artiba, and S. Iassinovski. An integrated production planning and scheduling system for hybrid flowshop organizations. *International Journal of Production Economics*, 74(1-3):33–48, 2001.

[159] R. Ruiz and C. Maroto. A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. *European Journal of Operational Research*, 169(3):781–800, 2006.

[160] R. Ruiz, F. S. Serifoglu, and T. Urlings. Modeling realistic hybrid flexible flowshop scheduling problems. *Computers & Operations Research*, 35(4):1151–1175, 2008.

[161] M. S. Salvador. A solution to a special case of flow shop scheduling problems. In S. E. Elmaghraby, editor, *Symposium of the Theory of Scheduling and Applications*, pages 83–91. Springer-Verlag, 1973.

[162] D. L. Santos, J. L. Hunsucker, and D. E. Deal. Flowmult: Permutation sequences for flow shops with multiple processors. *Journal of Information & Optimization Sciences*, 16(2):351–366, 1995.

[163] D. L. Santos, J. L. Hunsucker, and D. E. Deal. Global lower bounds for flow shops with multiple processors. *European Journal of Operational Research*, 80(1):112–120, 1995.

[164] D. L. Santos, J. L. Hunsucker, and D. E. Deal. An evaluation of sequencing heuristics in flow shops with multiple processors. *Computers & Industrial Engineering*, 30(4):681–691, 1996.

[165] D. L. Santos, J. L. Hunsucker, and D. E. Deal. On makespan improvement in flow shops with multiple processors. *Production Planning & Control*, 12(3):283–295, 2001.

[166] T. Sawik. Integer programming approach to reactive scheduling in make-to-order manufacturing. *Mathematical and Computer Modelling*, 46(11-12):1373–1387, 2007.

[167] T. J. Sawik. A scheduling algorithm for flexible flow lines with limited intermediate buffers. *Applied Stochastic Models and Data Analysis*, 9(2):127–138, 1993.

[168] T. J. Sawik. Scheduling flexible flow lines with no in-process buffers. *International Journal of Production Research*, 33(5):1357–1367, 1995.

[169] T. J. Sawik. Mixed integer programming for scheduling flexible flow lines with limited intermediate buffers. *Mathematical and Computer Modelling*, 31(13):39–52, 2000.

[170] T. J. Sawik. Mixed integer programming for scheduling surface mount technology lines. *International Journal of Production Research*, 39(14):3219–3235, 2001.

[171] T. J. Sawik. An exact approach for batch scheduling in flexible flow lines with limited intermediate buffers. *Mathematical and Computer Modelling*, 36(4-5):461–471, 2002.

[172] T. J. Sawik. Integer programming approach to production scheduling for make-to-order manufacturing. *Mathematical and Computer Modelling*, 41(1):99–118, 2005.

[173] T. J. Sawik, A. Schaller, and T. M. Tirpak. Scheduling of printed wiring board assembly in surface mount technology lines. *Journal of Electronics Manufacturing*, 11(1):1–17, 2002.

[174] P. Schuurman and G. J. Woeginger. A polynomial time approximation scheme for the two-stage multiprocessor flow shop problem. *Theoretical Computer Science*, 237(1-2):105–122, 2000.

[175] F. S. Serifoglu and G. Ulusoy. Multiprocessor task scheduling in multistage hybrid flow-shops: A genetic algorithm approach. *Journal of the Operational Research Society*, 55(5):504–512, 2004.

[176] S. V. Sevastianov. Geometrical heuristics for multiprocessor flowshop scheduling with uniform machines at each stage. *Journal of Scheduling*, 5(3):205–225, 2002.

[177] H. D. Sherali, S. C. Sarin, and M. S. Kodialam. Models and algorithms for a two-stage production

process. *Production Planning & Control*, 1(1):27–39, 1990.

[178] D. F. Shiau, S. C. Cheng, and Y. M. Huang. Proportionate flexible flow shop scheduling via a hybrid constructive genetic algorithm. *Expert Systems with Applications*, 34(2):1133–1143, 2008.

[179] V. Sirikrai and P. Yenradee. Modified drum-buffer-rope scheduling mechanism for a non-identical parallel machine flow shop with processing-time variation. *International Journal of Production Research*, 44(17):3509–3531, 2006.

[180] H. Soewandi and S. E. Elmaghraby. Sequencing three-stage flexible flowshops with identical machines to minimize makespan. *IIE Transactions*, 33(11):985–993, 2001.

[181] H. Soewandi and S. E. Elmaghraby. Sequencing on two-stage hybrid flowshops with uniform machines to minimize makespan. *IIE Transactions*, 35(5):467–477, 2003.

[182] C. Sriskandarajah and S. P. Sethi. Scheduling algorithms for flexible flowshops: Worst and average case performance. *European Journal of Operational Research*, 43(2):143–160, 1989.

[183] P. S. Sundararaghavan, A. S. Kunnathur, and I. Viswanathan. Minimizing makespan in parallel flowshops. *Journal of the Operational Research Society*, 48(8):834–842, 1997.

[184] C. S. Sung and H. A. Kim. A two-stage multiple-machine assembly scheduling problem for minimizing sum of completion times. *International Journal of Production Economics*, 113(2):1038–1048, 2008.

[185] V. Suresh. A note on scheduling of two-stage flow shop with multiple processors. *International Journal of Production Economics*, 49(1):77–82, 1997.

[186] K. Takaku and K. Yura. Online scheduling aiming to satisfy due date for flexible flow shops. *JSME International Journal Series C-Mechanical Systems Machine Elements and Manufacturing*, 48(1):21–25, 2005.

[187] L. X. Tang, W. X. Liu, and J. Y. Liu. A neural network model and algorithm for the hybrid flow shop scheduling problem in a dynamic environment. *Journal of Intelligent Manufacturing*, 16(3):361–370, 2005.

[188] L. X. Tang and H. Xuan. Lagrangian relaxation algorithms for real-time hybrid flowshop scheduling with finite intermediate buffers. *Journal of the Operational Research Society*, 57(3):316–324, 2006.

[189] L. X. Tang, H. Xuan, and J. Y. Liu. A new lagrangian relaxation algorithm for hybrid flowshop scheduling to minimize total weighted completion time. *Computers & Operations Research*, 33(11):3344–3359, 2006.

[190] L. X. Tang, Y. Zhang, J. Wang, X. Liao, and Z. Yi. Heuristic combined artificial neural networks to schedule hybrid flow shop with sequence dependent setup times. In J. Wang, X. Liao, and Z. Yi, editors, *Advances in Neural Networks - ISNN 2005. Second International Symposium on Neural Networks*, volume 3496 of *Lecture Notes in Computer Science*, pages 788–793. Springer Berlin / Heidelberg, 2005.

[191] R. Tavakkoli-Moghaddam, N. Safaei, and F. Sassani. A memetic algorithm for the flexible flow line scheduling problem with processor blocking. *Computers & Operations Research*, 36(2):402–414, 2009.

[192] H. W. Thornton and J. L. Hunsucker. A new heuristic for minimal makespan in flow shops with multiple processors and no intermediate storage. *European Journal of Operational Research*, 152(1):96–114, 2004.

[193] V. T'kindt and J.-C. Billaut. *Multicriteria scheduling: Theory, models and algorithms.* Springer, Berlin, second edition, 2006.

[194] A. D. Toni and S. Tonchia. Manufacturing flexibility: a literature review. *International Journal of Production Research*, 36(6):1587–1617, 1998.

[195] C. T. Tseng and C. J. Liao. A particle swarm optimization algorithm for hybrid flow-shop scheduling with multiprocessor tasks. *International Journal of Production Research*, 46(17):4655–4670, 2008.

[196] C. T. Tseng, C. J. Liao, and T. X. Liao. A note on two-stage hybrid flowshop scheduling with missing operations. *Computers & Industrial Engineering*, 54(3):695–704, 2008.

[197] H. Tsubone, M. Ohba, H. Takamuki, and Y. Miyake. A production scheduling system for a hybrid flow-shop: A case-study. *Omega-International Journal of Management Science*, 21(2):205–214, 1993.

[198] H. Tsubone, M. Ohba, and T. Uetake. The impact of lot sizing and sequencing on manufacturing performance in a two-stage hybrid flow shop. *International Journal of Production Research*, 34(11):3037–3053, 1996.

[199] H. Tsubone, M. Suzuki, T. Uetake, and M. Ohba. A comparison between basic cyclic scheduling and variable cyclic scheduling in a two-stage hybrid flow shop. *Decision Sciences*, 31(1):197–222, 2000.

[200] T. Uetake, H. Tsubone, and M. Ohba. A production scheduling system in a hybrid flow shop. *International Journal of Production Economics*, 41(1-3):395–398, 1995.

[201] G. Vairaktarakis and M. Elhafsi. The use of flowlines to simplify routing complexity in two-stage flowshops. *IIE Transactions*, 32(8):687–699, 2000.

[202] A. Vandevelde, H. Hoogeveen, C. Hurkens, and J. K. Lenstra. Lower bounds for the head-body-tail problem on parallel machines: A computational study of the multiprocessor flow shop. *Informs Journal on Computing*, 17(3):305–320, 2005.

[203] S. K. Verma and M. I. Dessouky. Multistage hybrid flowshop scheduling with identical jobs and uniform parallel machines. *Journal of Scheduling*, 2:135–150, 1999.

[204] A. Vignier, J.-C. Billaut, and C. Proust. Les problèmes d'ordonnancement de type flow-shop hybride : état de l'art. *RAIRO Recherche Opérationnelle*, 33(2):117–183, 1999.

[205] A. Vignier, J.-C. Billaut, C. Proust, and V. T'Kindt. Resolution of some 2-stage hybrid flowshop scheduling problems. In *1996 IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 2934–2941. IEEE Press, 1996.

[206] A. Vignier, P. Commandeur, and C. Proust. New lower bound for the hybrid flowshop scheduling problem. In *Proceedings of the 6th International Conference on Emerging Technologies and Factory Automation. ETFA '97*, pages 446–451. IEEE press, 1997.

[207] A. Vignier, D. Dardilhac, D. Dezalay, and C. Proust. A branch and bound approach to minimize the total completion timein a k-stage hybrid flowshop. In *Proceedings of the 5th International Conference on Emerging Technologies and Factory Automation. ETFA '96*, pages 215–220. IEEE press, 1996.

[208] S. Voss. The two-stage hybrid flow shop with sequence-dependent setup times. In A. Jones, G. Fandel, and T. R. Gulledge, editors, *Operations Research in Production Planning and Control. Proceedings of a Joint German/US Conference.*, pages 336–352. Springer-Verlag, 1993.

[209] S. Voss and A. Witt. Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements: A real-world application. *International Journal of Production Economics*, 105(2):445–458, 2007.

[210] J. A. Vázquez Rodríguez and A. Salhi. Performance of single stage representation genetic algorithms in scheduling flexible flow shops. In *The 2005 IEEE Congress on Evolutionary Computation*, volume 2, pages 1364–1371. IEEE press, 2005.

[211] J. A. Vázquez Rodríguez and A. Salhi. A synergy exploiting evolutionary approach to complex scheduling problems. In I. D. L. Bogle and J. Žilinskas, editors, *Computer Aided Methods in Optimal Design and Operations*, Series on Computers and Operations Research, pages 59–68. World Scientific Publishing Co. Pte. Ltd., 2006.

[212] J. A. Vázquez Rodríguez and A. Salhi. A robust meta-hyper-heuristic approach to hybrid flow shop scheduling. In K. P. Dahal, K. C. Tan, and P. I. Cowling, editors, *Evolutionary Scheduling*, Studies in Computational Intelligence, book chapter 5, pages 125–142. Springer-Verlag, Berlin Heidelberg, 2007.

[213] H. Wang. Flexible flow shop scheduling: optimum, heuristics and artificial intelligence solutions. *Expert Systems*, 22(2):78–85, 2005.

[214] H. Wang, V. Jacob, and E. Rolland. Design of efficient hybrid neural networks for flexible flow shop scheduling. *Expert Systems*, 20(4):208–231, 2003.

[215] X. Wang and L. Tang. A tabu search heuristic for the hybrid flowshop scheduling with finite intermediate buffers. *Computers & Operations Research*, 36(3):907–918, 2009.

[216] B. Wardono and Y. Fathi. A tabu search algorithm for the multi-stage parallel machine problem with limited buffer capacities. *European Journal of Operational Research*, 155(2):380–401, 2004.

[217] R. J. Wittrock. Scheduling algorithms for flexible flow lines. *IBM Journal of Research and Development*, 29(4):401–412, 1985.

[218] R. J. Wittrock. An adaptable scheduling algorithm for flexible flow lines. *Operations Research*, 36(3):445–453, 1988.

[219] Y. Wu, M. Liu, and C. Wu. A genetic algorithm for solving flow shop scheduling problems with

parallel machines and special procedure constraints. In *Proceedings of the Second International Conference on Machine Learning and Cybernetics*, volume 3, pages 1774–1779. IEEE press, 2003.

[220] W. Xiao, P. Hao, S. Zhang, and X. Xu. Hybrid flow shop scheduling using genetic algorithms. In *Proceedings of the 3rd World Congress on Intelligent Control and Automation*, pages 537–541. IEEE press, 2000.

[221] J. X. Xie and X. J. Wang. Complexity and algorithms for two-stage flexible flowshop scheduling with availability constraints. *Computers & Mathematics with Applications*, 50(10-12):1629–1638, 2005.

[222] J. X. Xie, W. X. Xing, Z. X. Liu, and J. F. Dong. Minimum deviation algorithm for two-stage no-wait flowshops with parallel machines. *Computers & Mathematics with Applications*, 47(12):1857–1863, 2004.

[223] H. Xuan and L. Tang. Scheduling a hybrid flowshop with batch production at the last stage. *Computers & Operations Research*, 34(9):2718–2733, 2007.

[224] T. Yang, Y. Kuo, and I. Chang. Tabu-search simulation optimization approach for flow-shop scheduling with multiple processors - a case study. *International Journal of Production Research*, 42(19):4015–4030, 2004.

[225] T. Yang, Y. Kuo, and C. Cho. A genetic algorithms simulation approach for the multi-attribute combinatorial dispatching decision problem. *European Journal of Operational Research*, 176(3):1859–1873, 2007.

[226] Y. Yang. *Optimization and heuristic algorithms for flexible flow shop scheduling.* PhD thesis, Columbia University, 1998.

[227] J. D. Yanney and W. Kuo. A practical approach to scheduling a multistage, multiprocessor flowshop problem. *International Journal of Production Research*, 27(10):1733–1742, 1989.

[228] V. Yaurima, L. Burtseva, and A. Tchernykh. Hybrid flowshop with unrelated machines, sequence-dependent setup time, availability constraints and limited buffers. *In press at Computers & Industrial Engineering*, 2009.

[229] K. C. Ying. An iterated greedy heuristic for multistage hybrid flowshop scheduling problems with multiprocessor tasks. *In press at Journal of the Operational Research Society*, 2009.

[230] K. C. Ying and S. W. Lin. Multiprocessor task scheduling in multistage hybrid flow-shops: an ant colony system approach. *International Journal of Production Research*, 44(16):3161–3177, 2006.

[231] K. C. Ying and S. W. Lin. Scheduling multistage hybrid flowshops with multiprocessor tasks by an effective heuristic. *In press at International Journal of Production Research*, 2009.

[232] M. Zandieh, S. M. T. Fatemi Ghomi, and S. M. Moattar Husseini. An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times. *Applied Mathematics and Computation*, 180(1):111–127, 2006.

[233] W. Zhang, C. Y. Yin, J. Y. Liu, and R. J. Linn. Multi-job lot streaming to minimize the mean completion time in m-1 hybrid flowshops. *International Journal of Production Economics*, 96(2):189–200, 2005.