Computers & Operations Research 70 (2016) 56–74

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Review

A comprehensive review of flowshop group scheduling literature

Janis S. Neufeld [a,*], Jatinder N.D. Gupta [b], Udo Buscher [a]

[a] Department of Business Administration and Economics, TU Dresden, 01069 Dresden, Germany
[b] College of Business Administration, University of Alabama in Huntsville, Huntsville, AL 35899, USA

ARTICLE INFO

Available online 23 December 2015

Keywords:
Flowshop
Group scheduling
Literature review
Family setup times
Cellular manufacturing

ABSTRACT

Due to its practical relevance, the flowshop group scheduling problem has received much attention in the academic and practice-oriented literature. As machines are grouped to cells and parts to part families, this problem is also known as cellular manufacturing scheduling. Group scheduling is characterized by sequencing tasks on two levels: on the one hand, a sequence of part families has to be determined considering major family setup times while, on the other hand, a job sequence has to be found within each part family. Despite an increasing number of publications, no comprehensive review on group scheduling and its solution methods has been conducted so far. This paper intends to close this gap by reviewing the development of research and characterizing the considered problem. All publications are categorized regarding the number of machines, type of setup times as well as the solution approach. Furthermore, group scheduling in flexible flowshop environments as well as a related scheduling task in multiple cells, known as cell scheduling problem, is considered. Finally, open problems and promising fields for future research in the area of flowshop group scheduling are identified.

Contents

* Corresponding author.
E-mail address: Janis_Sebastian.Neufeld@tu-dresden.de (J.S. Neufeld).

http://dx.doi.org/10.1016/j.cor.2015.12.006
0305-0548/

1. Introduction

The development and utilization of group technology (GT) and cellular manufacturing (CM) continue to have a great impact on the implementation of efficient batch manufacturing systems that face today's challenges [1]. While group technology is defined as "a manufacturing philosophy that identifies and exploits the underlying sameness of parts and manufacturing processes" [2], the implementation of GT in a manufacturing environment is referred to as CM [3], which has been GT's major application [4]. In cellular manufacturing systems (CMS), resources are divided into smaller organizational units called manufacturing cells, each of which produces certain sets of products referred to as part families. Besides the minimization of setup costs, the main advantage of CM is a simplification of material flows. Through the responsibility of a team for a limited set of parts, production control can be handled within each cell autonomously. As a result, an improved operator expertise is achieved, which leads to a more reliable production with improved quality and lower rework costs. Furthermore, implementation of GT and CMS can achieve shorter throughput times, lower stocks, decreases material handling and production costs [4,5]. Thus, CM is especially advantageous for systems with complex material flows and a high level of automation like flexible manufacturing systems.

Three major planning tasks become necessary in CMS, which, apart from few exceptions [6,7], are usually solved independently: First, the cell formation problem which implies the grouping of machines to manufacturing cells as well as the formation of part families and their assignment to cells. Part families are usually formed according to the required operations, machines and tooling. Second, a layout problem has to be solved by positioning manufacturing cells in the shop floor and machines within each cell. Finally, all jobs and operations have to be scheduled. The cell formation and cell layout problem have received much attention in the literature [3,8,9]. However, research on developing efficient and effective scheduling systems that are crucial to fully gain the advantages of CM has appeared only recently. While traditional production systems often constitute job shop environments, CMS flowline manufacturing cells are quite common in practice, as the objective of GT is to create simple material flows [10]. Either the assignment of more than just one part family to a cell or the splitting of part families to subfamilies that require a similar tooling (also "tooling families") still results in a complex sequencing task referred to as group scheduling: on the one hand, a sequence of jobs or parts within each part family has to be determined while, on the other hand, a family sequence must be identified taking family setup times into account. While these two decisions are described separately, there is an interaction and interdependence between them, thus, increasing the complexity of formulating and solving the resulting group scheduling problem.

Group scheduling has proven to be relevant in various areas beyond classical CMS even though the literature mostly uses terminology and developments of CMSs. This is shown by its diverse practical applications, such as automotive paint and body shops [11], furniture production [12], label sticker manufacturing [13], semiconductor industry [14], blades for airplane engines [15], punching machines for metal parts [16], manufacturing of centrifugal pumps [17], printed circuit board (PCB) [18], TFT-LCD production [19] or electronics manufacturing in general [20] but also bridge construction problems [21] and wine bottling [22]. After Hitomi and Ham [23] first stated the flowshop group scheduling problem in 1976, the development of algorithms to solve this problem has attracted numerous researchers and practitioners. Vaithianathan and McRoberts [24] discussed the idea of decomposing scheduling problems with part families into two levels and compared the complexity of the arising problem to traditional flowshop and job shop problems. Following this, through simulation studies in dynamic environments with stochastic job arrivals, it was shown that exhaustive scheduling rules, i.e. all parts of a family are processed subsequently, outperform non-exhaustive rules in terms of both, mean flow time and mean tardiness [25]. After that, especially algorithms that were originally developed to solve traditional flowshop problems have been modified and applied to solve static group scheduling problems, where all jobs, part families and parameters are known in advance. While in the beginning sequence-independent setup times have been considered only, Flynn [26] highlighted the need to include sequence-dependent setups, which has been the predominantly considered type of setup time recently.

Despite strong computational resources nowadays, only small to medium size problems can be solved by exact methods [20]. Hence, various constructive heuristics and metaheuristic approaches have been developed for the real-world flowshop group scheduling problems. Nevertheless, even though various reviews on scheduling had been conducted within the last decades, until today no comprehensive overview of group scheduling approaches has been given. Among these were studies covering scheduling problems with setup considerations and, hence, partly containing group scheduling literature. Allahverdi et al. [27,28] and Allahverdi [29] presented thorough reviews on scheduling problems with setup considerations in various shop environments, whereas other surveys confined themselves to flowshop scheduling in general [30] or scheduling problems with sequence-dependent setup times [31,32]. Potts and Kavalyov [33] concentrated on scheduling with batching, of which group scheduling can be seen as a special case of serial batching. However, due to the exponentially growing number of publications on scheduling, there is still a recurring need for up-to-date surveys focusing on either specific solutions methods or particular problem classes [28]. Therefore, in this paper, we provide a comprehensive review of the development of the approaches and algorithms developed to solve the flowshop group scheduling problems excluding flowshop group scheduling of identical jobs (also known as batch scheduling of identical jobs) and the class of lot-streaming problems. To the best of our knowledge, this is the first approach to systematically give an overview of flowshop group scheduling literature and to provide guidance for a classification of existing solution algorithms.

The rest of this paper is structured as follows: First, a detailed definition of the flowshop group scheduling problem, its extensions and complexity as well as commonly used solution representations are presented in Section 2. Afterwards, in Section 3 a distinction to related problems and an analysis of flowshop group scheduling approaches is conducted, which emphasizes the development of literature and its solution approaches over the last years and identifies the relevant journals for publications on group scheduling. The specific publications are classified and presented into four categories following the historical development of group scheduling literature. Section 4 summarizes simulations studies for solving the group scheduling problems in dynamic environments, that were prevalent in the early 1990s in order to validate the potential of CM. Since most publications on the static flowshop group scheduling problem have focused on scheduling several part families within an autonomous manufacturing cell, Section 5 separately presents the approaches for two-machine flowshop problems in Section 5.1 and multiple machine flowshops in Section 5.2. In this section, the publications are structured according to sequence-independent and sequence-dependent type of setup times as well as the considered optimization criteria. Group scheduling problems in flexible flowshop environments with parallel machines on at least one stage are discussed in Section 6. Section 7 focuses on solution algorithms for scheduling multiple cells, that are usually modeled as flowshop group scheduling problems. Finally, Section 8 concludes the paper with some fruitful directions for future research.

2. Classification, problem specification and complexity

2.1. Basic group scheduling problem

The classical static flowshop group scheduling problem is defined as follows: consider a given set of n jobs, $\Omega = \{1, 2, \ldots, n\}$, which is to be processed on a set of m machines, $\Psi = \{1, 2, \ldots, m\}$, in the same technological order. Every job belongs to a certain part family from the set of part families $\Phi = \{1, 2, \ldots, F\}$. Let n_f represent the number of jobs assigned to a family f and $\Omega_f = \{1, 2, \ldots, n_f\}$ the respective set of jobs. Each job may be available for processing on the first machine at an individual release time $r_j \geq 0$, even though in literature usually all jobs are available at the beginning of the planning horizon. A sequence-independent (or sequence-dependent) family setup time s^i_f ($s^i_{f',f}$) has to be taken into account for every changeover (from a part family f') to family f on machine i. For the case of sequence-dependent setup times an initial setup time $s^i_{0,f}$ occurs if family f is at the first position of the family sequence. Setup times between two jobs of the same family are negligible or can be included in the processing times. Generally, two types of setup times are identified: anticipatory setup times (also detached) that can be carried out as soon as the respective machine is idle and, in contrast, non-anticipatory setups that are attached to the jobs and can be started only after arrival of the job at a machine. While sequence-dependent setup times are anticipatory by their very nature [30], sequence-independent, non-anticipatory setup times can often be integrated into the processing times of the jobs. However, for family setup times it is not possible to be included into the processing time of a certain job, since the first job in the sequence of a family is not known before the scheduling problem is solved. Hence, most scheduling research focuses on anticipatory setups [30]. Furthermore, arising in PCB manufacturing, carryover-sequence-dependent setup times are considered occasionally, i.e. setups are not only dependent on the immediately preceding family but *all* preceding families [20].

A fundamental characteristic of group scheduling is the so-called group technology assumption, stating that all jobs of a part family must be processed consecutively without interruption by jobs of a different part family or being split into sublots (also called exhaustive scheduling). Thus, in flowshop environments two levels of scheduling problems, that are closely related, can be identified. At the first level, a sequence of jobs (or parts) within each part family has to be determined, which is called a job sequence. At the second level, a family sequence is identified, preferably an optimal sequence of part families or the so-called tooling families (subfamilies) [34]. Even though these two levels of scheduling can be described separately, there is an interdependence and interactions appear between these two levels while solving the problems. Together, the family sequence and all job sequences form a group schedule. Hence, in permutation flowshops the number of sequences is limited to a single family sequence and n_f job sequences, while considering non-permutation schedules in total $m \cdot n_f$ job sequences and m family sequences can be considered. Due to the increasing degree of complexity, most commonly permutation flowshops were regarded only. Lately, additional characteristics such as no-wait restrictions or limited buffers have been integrated in order to meet the conditions of practical scheduling applications.

Often the considered flowshop group scheduling problem is referred to as scheduling in (pure) flowline manufacturing cells, even though some researchers point out a significant difference: Rajendran [35] defines flowline scheduling as environment in which certain jobs may skip some stages, also referred to as missing operations, while in flowshop scheduling all jobs should be processed on all stages. In contrast, in the majority of publications both terms are not differentiated (e.g. [18,36,37]) or in flowline manufacturing it is explicitly defined that jobs have to visit all stages (e.g. [38,39]). Hence, in this review the terms flowshop and flowline are used synonymously and publications that explicitly consider missing operations are highlighted.

Except for the two-machine makespan flowshop scheduling problem, all static flowshop scheduling problems with $m \geq 2$ are known to be NP-hard in the strong sense for all studied optimality criteria [40–42]. As the general flowshop is a special case of flowshop scheduling with setup times, the group scheduling problems are also NP-hard in the strong sense since the static two-machine flowshop group scheduling problem with sequence-dependent setup times to minimize makespan has been shown to be NP-hard in the strong sense [43,44]. Hence, in general it is possible to solve group scheduling problems optimally for small problem instances only and (meta)heuristic approaches are used most commonly to find approximately optimal solutions for the large-sized problems.

Besides static and deterministic problems, where all parameters are fixed and known in advance, flowshop group scheduling has also been considered in dynamic environments. In dynamic environments job arrivals follow a stochastic distribution and cannot be foreseen.

2.2. Extensions of the basic group scheduling problem

Nowadays, in many companies machines in some stages are duplicated in order to increase capacity or add flexibility to a manufacturing system. The resulting scheduling problem differs from the basic flowshop group scheduling problem: Again a given set of n jobs, $\Omega = \{1, 2, \ldots, n\}$, is considered, which now has to be processed on c stages, $\overline{\Psi} = \{1, 2, \ldots, c\}$, in the same technological order. At least one stage consists of more than one parallel machines. These parallel machines can either be identical, uniform or unrelated. While uniform machines are characterized by a specific associated speed v_i, resulting in machine-dependent processing times of p_{ij}/v_i, processing times on unrelated parallel machines are individual on each machine and do not relate to each other. In this case, some machines may even not be able to process all jobs, which is referred to as machine eligibility restriction. Again, every job belongs to a certain part family from the set of part families $\Phi = \{1, 2, \ldots, F\}$ and a sequence-independent (or sequence-dependent) family setup time s^i_f($s^i_{f',f}$) has to be taken into account for every changeover (from a part family f') to family f on machine i. The existence of parallel machines at one or several stages generally results in a more complex scheduling problem, since besides a sequencing task of jobs and part families additionally jobs or part families have to be assigned to one of the parallel machines. Similar to the basic group scheduling problem, the group technology assumption requires that all jobs of a part family have to be processed successively. However, it may be beneficial to assign parts from the same family to several parallel machines, each of which still keeping the group technology assumption. In this case multiple setups become necessary. If this splitting of part families is not permissible, only single setups per part family occur [45]. The flowshop group scheduling problem with parallel machines at some stages is referred to as hybrid flowshop, flexible flowshop or multiprocessor flowshop group scheduling problem. If certain jobs or part families can skip certain stages, the problem is sometimes called a hybrid flexible flowshop group scheduling problem (e.g. [19,46,47]).

Besides the classical case of flowshop group scheduling, which typically arises in a single autonomous cell, the relevance of considering material flows between different manufacturing cells, the so-called intercellular movement, has been pointed out [48]. As in many practical cases it is not possible to form completely independent cells, the processing of parts that have originally been assigned to other cells (exceptional elements) has great impact on material flows and, therewith, the scheduling task in CMS: while usually the majority of operations has to be performed in a part's original cell, exceptional elements have at least one operation that is processed on a machine in a different cell and, hence, need to be transported between the two cells. Machines that are assigned to a certain cell but still are required for processing parts from different manufacturing cells are called bottleneck machines. Even though scheduling multiple cells generally cannot be classified as flowshop scheduling, Solimanpur et al. [49] defined the consideration of flowline manufacturing cells as *cell scheduling problem* that is modeled and solved similar to the group scheduling problem. Assuming the assignment of a single part family to each cell only, again, a scheduling task on two levels has to be solved: On the one hand, a sequence of jobs within each manufacturing cell (or family) has to be determined (intra-cell scheduling), while, on the other hand, the order of processing part families belonging to a certain cell as well as exceptional elements is resolved by a shop-wide cell sequence (or family sequence), which is referred to as inter-cell scheduling. For every changeover between jobs, which are assigned to different cells, the necessity of either sequence-independent or sequence-dependent setup times arises.

2.3. Notation

For a precise classification of the considered static problems we follow the common three-field notation $\alpha|\beta|\gamma$ introduced by Graham et al. [50] and extended by Pinedo [51]. Relevant parameters are presented in Table 1, that also gives an overview of all considered problem attributes and objectives in flowshop group scheduling literature. In contrast to basic flowshop problems with m machines denoted as $\alpha = Fm$, following Ribas et al. [52], α is disaggregated into additional parts $FFc(\alpha_1\alpha_2^{(1)}, \alpha_1\alpha_2^{(2)}, \ldots, \alpha_1\alpha_2^{(c)})$ for flexible flowshop scheduling problems. While FFc indicates the existence of parallel machines in at least one of c stages, α_1 and α_2 specify the characteristics for each stage $1, \ldots, c$. α_1 gives information whether the machines are identical (P), uniform (Q) or unrelated (R) and α_2 indicates the number of machines at the respective stage. Besides, Ribas et al. [52] propose a grouping of consecutive stages if these are equal, with $((\alpha_1\alpha_2)^l)_{l=s}^k$ representing $k-s$ similar consecutive stages. For example, $FF3, (0, (P4^l)_{l=2}^3)$ indicates a flexible flowshop with 3 stages where there are no parallel machines in the first stage and four identical parallel machines in the second and third stage. In addition to the notation introduced previously, two novel terms for β are suggested in order to describe the considered problems completely: $\beta = gta$ denotes that the group technology assumption is regarded. This implies the existence of part families commonly referred to as *fmls*. $\beta = cs$ characterizes the existence of intercellular moves, resulting in the cell scheduling problem.

Table 1
Problem characteristics for three-field notation.

α	
Fm	Flowshop with m machines
$FFc(\ldots)$	Flexible flowshop with c stages
Pm^c	m identical parallel machines at stage c
Qm^c	m uniform parallel machines at stage c
Rm^c	m unrelated parallel machines at stage c
β	
s_f	Sequence-independent family setup times
s_{fg}	Sequence-dependent family setup times
r_j	Release dates
gta	Part families with group technology assumption
fmls	Part families
prmu	Permutation flowshop
nwt	No-wait flowshop
prmp	Preemption of jobs allowed
block	Limited buffers
cs	Cell scheduling problem with exceptional elements
M_j	Machine eligibility restrictions
t_j	Removal / transportation times
unavail	Machine availability restrictions
other	Other problem characteristics
γ	
C_{max}	Makespan
$\sum C_j$	Total completion time
$\sum F_j$	Total flow time
$\sum w_j C_j$	Total weighted completion time
$\sum w_j F_j$	Total weighted flow time
$\sum T_j$	Total tardiness
$\sum w_j T_j$	Total weighted tardiness
T_{max}	Maximum tardiness
L_{max}	Maximum lateness
EE	Number of exceptional elements
cost	Multi-objective cost function

$$\left(\underbrace{3,1,4,2}_{\substack{\textit{family}\\ \textit{sequence}}} \;\Big\|\; \underbrace{3,1,4,2}_{\substack{\textit{job seq.}\\ \textit{family 1}}} \;\Big|\; \underbrace{5,1,3,2,4}_{\substack{\textit{job seq.}\\ \textit{family 2}}} \;\Big|\; \underbrace{1,4,2,3}_{\substack{\textit{job seq.}\\ \textit{family 3}}} \;\Big|\; \underbrace{3,1,2}_{\substack{\textit{job seq.}\\ \textit{family 4}}} \right)$$

Fig. 1. Solution representation for a permutation flowshop group scheduling problem.

2.4. Problem representation

In order to solve the flowshop group scheduling problem with constructive and metaheuristic approaches, a common solution representation evolved, that was introduced by Franca et al. [53] originally. A sequence σ (group schedule) is divided into n_f+1 sections, each of which representing a family or job sequence. While the first section stands for the family sequence σ_F containing all part families, the second section represents the first job sequence σ_{J1}, the third one represents the second job sequence σ_{J2} and so on (see example in Fig. 1). This solution representation has the advantage that changes in the family and job sequences do not affect the representation of each other and both levels can be solved independently to find an approximately optimal schedule. While this encoding scheme is originally designed to represent permutation schedules only, it can easily be extended for non-permutation schedules. Lin et al. [54] allow differing family sequences on every machine. Hence, the number of family sequences is extended to m, one for each machine, while the sequence of all jobs within each family remains constant. Therefore, a group schedule is represented by in total n_f+m sections.

Non-permutation schedules within the job sequences of each family have been considered by Nikjo and Zarook [55] only. They multiply the solution representation displayed in Fig. 1 by the number of machines in order to describe an individual family sequence and individual job sequences on each machine. This results in $m(n_f+1)$ separate sequences and leads to a very complex problem structure.

Chen et al. [56,57] discuss a new representation based on a single permutation of all jobs regardless of their family. The family sequence is determined by the position of the first job of each family in this sequence. The new encoding scheme is compared to the traditional approach developing two taboo search algorithms [56] as well as two ant colony optimization and genetic algorithm approaches [57]. In general, the algorithms applying new problem representations solve benchmark instances more effectively, while, at the same time, requiring significantly higher computation times. Besides, the novel algorithms have not been compared to existing lower bounds or state of the art metaheuristic algorithms. Nevertheless, results indicate that the use and development of different solution representations should be further investigated. Another similar, novel encoding scheme for the two-machine permutation flowshop was discussed by Liou et al. [58], who use a single permutation for all jobs regardless of their family together with an additional group vector. These vectors have to be transformed to a feasible group schedule in the course of the optimization algorithm. Similar to those by Chen et al. [56,57], the proposed representation neglects the jobs' affiliations to part families in the first place and, hence, different solutions may lead to the same group schedule. This encoding scheme is also used for solving m-machine flowshop group scheduling problems by Liou and Hsieh [59]. Another alternative modeling for the two-machine permutation problem is presented by Logendran et al. [60]. On the basis of the structure of a single family schedule, the sequencing of families is solved as a traveling salesman problem. However, this approach did not receive further attention and has not been compared to other solution representations.

In addition to job and family sequences, flexible flowshop group scheduling problems require an assignment of jobs or part families to parallel machines. In the literature several approaches evolved to represent a solution for this problem. In most cases, the classical decoding scheme from the basic group scheduling problem was used, assigning families to the parallel machine that enables the earliest possible start time of the first job of the respective family (e.g. [45,61–63]). Especially for problems with identical parallel machines this is reasonable since the solution representation remains rather simple. Based on Norman and Bean [64], Zandieh et al. [65] introduce a different matrix representation incorporating the idea of random key genetic algorithms, where random numbers are used for coding solutions. For each part family a random number is generated, whose integer part determines the machine it is assigned to whereas the fractional part is used for sequencing the families. For each job a fractional part is generated only intended for the first level of group scheduling. This solution representation is also used in several other publications (e.g. [66–68]). Another decoding scheme is proposed by Bozorgirad et al. [69], who depending on different algorithms use two or three different matrices: First, a three-dimensional jobs-matrix represents the job sequences within each family at all stages. Second, either a three-dimensional groups-matrix determines the assignment of part families to machines on all stages as well as the sequence of families on each machine or a two-dimensional machines-matrix represents the assignment of families to machines at all stages together with a one-dimensional groups-matrix that constitutes a sequence of families which must be followed in all stages.

3. Literature overview and analysis

3.1. Limitations of research

In order to identify all relevant publications in flowshop group scheduling problems, in April 2014 as well as December 2014, we searched several scientific databases like “Science Direct”, “Business Source Complete” and “Google Scholar” using combinations of the specifying terms “group scheduling”, “family setup times” and “flowshop/flowline manufacturing cell”. The results were screened according to the problem definition of flowshop group scheduling specified above. Furthermore, in group scheduling literature all cited references as well as referring papers were examined. Even though the aforementioned specification of the flowshop group scheduling problem is well established, some existing research studies use the term “group scheduling” or “scheduling in flowline manufacturing cells” while discussing a variety of other problems. Hence, we establish the boundary of the literature incorporated in this paper as follows:

- Several papers have been published on scheduling in manufacturing cells even though solely a sequence of jobs within a single part family is determined [35,70–76]. Missing operations, that are typically found in CM, are explicitly taken into account. However, if the first scheduling level is only considered, the problem is reduced and solved as an ordinary flowshop [34]. Other researchers have solely focused on sequencing part families, i.e. the second level of group scheduling [77,78]. Nevertheless, the interaction between both levels is a significant characteristic and is crucial for an evaluation of solution methods. Therefore, papers considering a single level of scheduling only are not included in this review.
- A more general problem is formulated if the group technology assumption is neglected and non-exhaustive scheduling of part families becomes permissible. This is the case for several dispatching rules used for simulation studies in dynamic environments as well as static problems [79–87]. However, the possibility of splitting part families in multiple batches adds another dimension, which is the assignment of jobs to batches. Hence, the resulting problem can be seen as serial batch scheduling with job availability and batch setup times. Alternatively, it can be transformed into an ordinary flowshop problem without part families but sequence-dependent setup times, as jobs belonging to the same part family do not have to be processed in succession. In both cases a more complex scheduling problem has to be solved. Hence, non-exhaustive scheduling as well as inconsistent families [88,89] will not be discussed in this paper.
- Scheduling problems with identical jobs within each part family, that have to be processed successively, are studied by Pranzo [90] and Yang et al. [21]. As identical jobs also obviate the need for sequencing the first level scheduling problem, these problems are also not included in our review below. Similarly, flowshop scheduling with setup times and lot-streaming or lot splitting considerations can bear resemblance to group scheduling (e.g. [91,92]). However, the formation of lots is usually based on identical jobs and excludes the group technology assumption and, hence, these approaches are not included in our discussion below.
- Few papers focus on a joint solution of several planning tasks in cellular manufacturing such as integrating cell formation, cell layout and scheduling problem [7] or machine loading and scheduling [6]. Due to a significant change of the problem's characteristics, these papers are not classified as group scheduling literature and hence are not considered in this review.

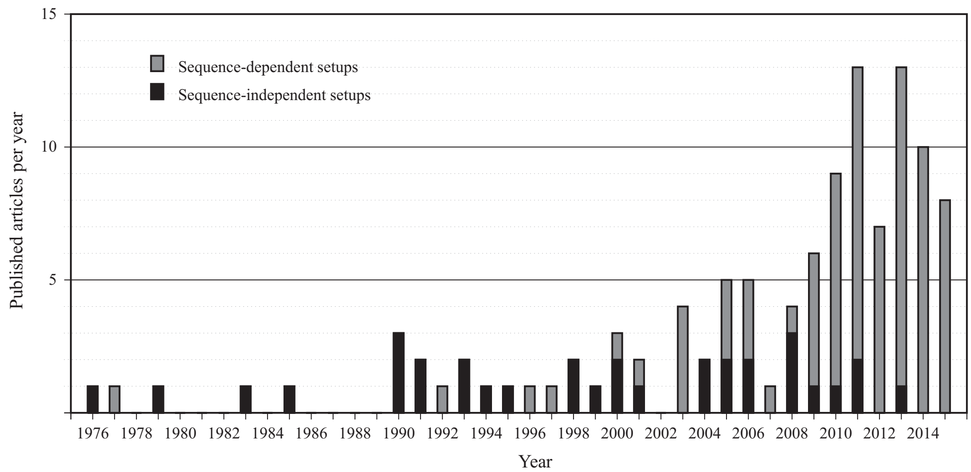

Fig. 2. Publications per year on flowshop group scheduling problems.

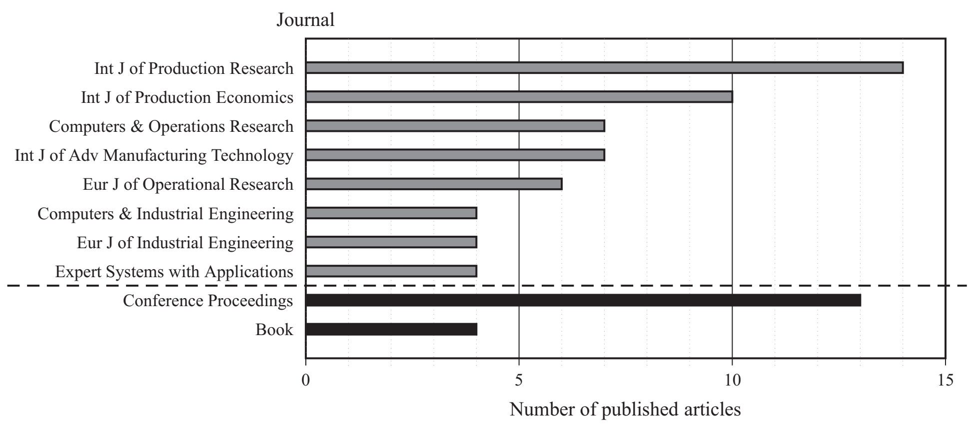

Fig. 3. Publications per journal (journals with more than three publications only).

3.2. Analysis of existing publications

In our review, in total 112 papers published have been identified focusing on the analysis and solution of the flowshop group scheduling problem. Among these 7 papers study dynamic environments, where mainly dispatching rules are applied, while the majority of 105 papers discusses static cases. Fig. 2 shows the development of publications per year, which is characterized by an increasing number of publications since the early 1990s. During the early years (starting around 1976), group scheduling literature focused on the consideration of sequence-independent setup times (in total 33 papers). Recently, problems with sequence-dependent setups have received greater attention (in total 79 papers). The most relevant journals concerning the number of publications on flowshop group scheduling are displayed in Fig. 3. Majority of research in group scheduling is published in *International Journal of Production Research*, *International Journal of Production Economics*, *Computers & Operations Research*, and *International Journal of Advanced Manufacturing Technology*.

The pursued optimization criteria are summarized in Table 1 in Section 2.3. For the total flow (or completion) time objective, some researchers minimize mean flow (completion) time. As these criteria lead to the same solution they are not reported separately. Furthermore, in nearly all static cases all jobs are considered to be available at the beginning of the planning horizon. With this, the objectives total flow time and total completion time lead to the same results. In order to emphasize the comparability of approaches developed for the same problem, all research considering these two objectives with $r_j = 0, \forall j$, is referred to as $\sum C_j$. Similar to the scheduling literature in general, in flowshop group scheduling, most frequently used objective is the minimization of makespan (66 publications). Besides, in 20 cases, flow time or completion time based criteria and in 20 cases, tardiness or lateness based criteria are minimized. Bi- or multi-criteria objective functions are considered by 21 articles.

3.3. Solution approaches for solving group scheduling problems

Concerning the solution methodology, all 7 publications on group scheduling in dynamic environments focus on the use of combined dispatching rules in order to efficiently generate solutions under constantly changing conditions. Only Wemmerlöv and Vakharia [93] consider simple constructive algorithms based on well-known procedures for flowshop problems, namely NEH [94] and CMD [95]. In contrast, various solution methods have been applied for solving static group scheduling problems. Using existing MIP formulations such as by Salmasi et al. [11], Lu and Logendran [96] or Sabouni and Logendran [97] multiple machine group scheduling problems with more than 5 families and 20 jobs can usually not be solved within computation times of several hours. Naderi and Salmasi [98], however, propose a very efficient mathematical model, that is able to solve problems with up to 60 jobs in a reasonable amount of time. This indicates that there is still room for improving exact formulations significantly. Nevertheless, due to the NP-hard nature of the problem, optimizing algorithms and MIP models are capable of solving small to medium size problems only. In total 28 publications present MIP models, even though most of these are based on a similar

Table 2
Simulations studies in dynamic group scheduling environments.

Publication	Setup type	Objectives: Flow time based	Objectives: Due date based	Promising rules (family rule / job rule): Flow time based	Promising rules (family rule / job rule): Due date based	Comment
Hitomi et al. [112]	s_{fg}	×	×			analysis of shop type, different flow patterns for each family
Russell and Philipoom [113]	s_f	×	×	APT/SPT	EDD/EDD, FCFS/EDD, FCFS/SL, FCFS/EDD	non-exhaustive and exhaustive rules
Wemmerlöv and Vakharia [93]	s_f	×	×	FCFC/FCFS	FCFC/FCFS	non-exhaustive and exhaustive rules
Mahmoodi et al. [114]	s_{fg}	×	×	MS/SPT	EDD/TSPT	non-exhaustive and exhaustive rules
Frazier [115]	s_{fg}	×	×	MJ/SPT	EDD/TSPT	
Reddy and Narendran [116]	s_{fg}	×	×	MJ/EDD,PH/SPT	MJ/SPT, PH/SPT	non-exhaustive and exhaustive rules
van der Zee et al. [17]	s_f	×		MASP_AD	–	flexible flowshop cell, non-exhaustive and exhaustive rules

s_{fg}: sequence-dependent setups; s_f: sequence-independent setups; APT: average processing time; SPT: shortest processing time; EDD: earliest due date; FCFS: first come first serve; SL: slack; MJ: most jobs; PH: predictive heuristic; MS: minimum setup time; TSPT: two class truncated SPT; MASP_AD: adaptive minimum ave. setup plus processing time.

formulation presented by Salmasi et al. [11,99]. Furthermore, 9 papers propose branch and bound algorithms while 5 variations of Johnson's rule [100] for solving two-machine problems optimally are discussed. Hence, the majority of publications focuses on heuristic and metaheuristic approaches. Among these, comparisons of several constructive algorithms are conducted 13 times, while 75 papers propose metaheuristic algorithms. Despite partly sophisticated procedures, the reported computation times remain reasonable and acceptable for real-world problems. For example, the most commonly used benchmark instances for the group scheduling problem with multiple machines and sequence-dependent setup times by Schaller et al. [18] use problem sizes with up to 10 machines, 10 part families and 10 jobs per part family. Even the largest of these instances efficient metaheuristics require less than 60 s, while the average computation time is measured with usually about 20 s (see e.g. [39,97,101]). Thus, up-to-date metaheuristics are capable of solving even large sized group scheduling problems within acceptable computation times using modern computer technology. At the same time, constructive heuristics, of course, require significantly lower computational effort of less than one second for all test instances still providing good solutions.

3.4. Existing benchmark instances

In order to prove and compare the effectiveness and efficiency of solution algorithms the existence of common benchmark instances is crucial. While for the basic static flowshop group scheduling problem with sequence-independent setup times as well as two-machine group scheduling problems no shared test beds have been proposed, two sets of test problems have been applied regularly for the case with sequence-dependent setups. Schaller et al. [18] developed in total 900 instances with at most 10 machines, 10 part families, and 10 jobs in each family. These are subdivided into three classes of setup times with setup to processing time ratios between approximately 2:1 and 10:1, while all processing times are uniformly distributed integers of the range $[1, 10]$. These instances are well established and used within 17 other publications. Salmasi et al. [11] attempt to develop more realistic instances. Since real-world manufacturing cells usually are rather small and, at the same time, may have more than 10 part families, they propose test instances with 2–6 machines and a maximum number of 16 families each of which consisting of up to 10 jobs. Furthermore, the ratio of setup times between consecutive machines is considered as a significant factor in generating different test problems. Hence, different levels of increasing, decreasing or constant setup to processing time ratios are considered. These test instances have also been used regularly in recent publications. For the flexible flowshop group scheduling problem there are no prevalent test instances in the literature, even though nearly all publications use similar problem structures based on those presented by Logendran et al. [102] and Zandieh et al. [65]. Generally, between 2 and 16 part families with up to 12 jobs are taken into account, while 2–9 stages comprise between a single and 4 parallel machines. Compared to benchmark instances for the basic flowshop group scheduling problem the setup to processing times ratio is often rather low, with processing times uniformly distributed in the range $[5, 75]$ and family setup times within $[5, 25]$ (e.g. [47,65,102]). Especially in flexible flowshop environments the influence of different setup and processing times has not been discussed exhaustively. Developed algorithms for the cell scheduling problem have been tested for a small number of 16 sample problems from the literature only, that have originally been developed to solve the cell formation problem (see [49,103]). Thus, there is still a need of developing benchmark instances that model the variety of real-world cellular manufacturing systems and take different material flows into account.

4. Dynamic group scheduling via simulation studies

In the early 1990s, primarily simulation studies in dynamic environments have been conducted in order to prove the effectiveness of CM and group scheduling approaches. The used simulation models are characterized by dynamic job arrivals following a stochastic distribution. Furthermore, other parameters, such as family membership, job processing times or required operations, are usually generated dynamically and not known in advance. These studies examine dispatching rules or simple two-stage heuristics to solve. In addition to a consideration of flowshop manufacturing cells, several simulation studies focused on job shop manufacturing cells [25,104–111]. Generally, the effectiveness of certain dispatching rules and the relevance of different influencing factors were similar for both shop environments.

All simulation studies with part families in flowline manufacturing cells are summarized in Table 2. The first simulation study that considers family-based dispatching rules, i.e. rules that take the group technology assumption into account, which is also referred to as exhaustive scheduling, was published by Hitomi et al. [112]. To minimize maximum flow time, they compared 8 different dispatching rules, from which one considers the existence of part families in different shop environments. Each group of jobs is characterized by a different flow pattern and sequence-dependent family setup times. However, they did not explicitly analyze the benefits of using family-based rules. The advantageousness of exhaustive scheduling compared to traditional non-exhaustive dispatching rules and heuristics was proven by Russell and Philipoom [113] and Wemmerlöv and Vakharia [93]. Russell and Philipoom [113] compared several known as well as novel dispatching rules, while Wemmerlöv and Vakharia [93] also tested modified simple heuristics based on CDS [95] and NEH [94]. Generally, marked improvements of family-based scheduling approaches could be identified for flow time as well as lateness-oriented measures. In both publications, sequence-independent setup times were considered.

Later simulation studies focused on the identification of superior dispatching rules for various criteria in cells with sequence-dependent setups. Mahmoodi et al. [114] analyzed the influence of varying experimental conditions like different utilization or distributions of job arrivals. Frazier [115] compared in a comprehensive study in total 14 scheduling rules from which the MJ/SPT rule performed best by selecting the part family with the maximum number of waiting jobs while within each part family the jobs are sequenced according to the shortest processing time. A novel heuristic for determining the family sequence was proposed by Reddy and Narendran [116] and showed a promising performance, especially for high levels of utilization. A special type of flexible flowshop manufacturing cell, with a batching machine at the first stage and parallel machines for each part family at the second stage, was studied by van der Zee et al. [17]. Several exhaustive and non-exhaustive dispatching rules were used for the batching machine in order to analyze their influence on the shops performance on succeeding stages concerning average flowtime and average waiting time. Thereby, family-based non-exhaustive scheduling rules show the best performance. Apart from the work by Reddy and Narendran [116] and van der Zee et al. [17], no other simulation studies in dynamic environments have been published in the literature since 1996, despite its close resemblance to many real-life manufacturing environments and, therewith, high relevance for practical applications.

5. Static group scheduling in a single cell

5.1. The two-machine flowshop

Table 3 gives an overview of publications on two-machine problems. The first publication on the two-machine flowshop group scheduling problem with sequence-independent setup times was provided by Yoshida and Hitomi [117] considering $F2|s_f, prmu, gta|C_{max}$ problem. A modification of Johnson's rule [100] was applied for generating optimal schedules concerning makespan. A further analysis was done by Sekiguchi [118]. Based on the definition of precedence relations, an algorithm was developed for solving the problem optimally. Ham et al. [2] described in their fundamental work the basic problem definition, several dominance conditions as well as algorithms for solving the group scheduling problem.

An analysis leading to a further understanding of the problem was presented by Baker [119], who modeled each part family as job with time lag. Again, modified Johnson's rule [100] was applied to solve $F2|s_f, prmu, gta|C_{max}$ problem optimally. However, the algorithm applied by Baker was proven not to generate an optimal solution on the first level of scheduling in all cases by Kleinau [44]. Yang and Chern [120] proposed a generalization of the algorithms by Johnson and Baker respectively. In addition to sequence-independent setup times, removal and transportation times were included in their problem definition, which was solved by a

Table 3
Two-machine flowshop group scheduling literature.

Publication	Setup type	Objective	Bi-/multi-criteria	MIP	Exact algorithm	Constr. algorithm	Meta-heuristic	Solution method	Comment
Yoshida and Hitomi [117]	s_f	C_{max}			×			Johnson	
Sekiguchi [118]	s_f	C_{max}			×			Series-parallel algorithm	
Ham et al. [2]	s_f	C_{max}			×	×		B&B, Johnson, constr.	
Baker [119]	s_f	C_{max}			×			Johnson	
Logendran [34]	s_f	C_{max}				×		constr.	
Yang and Chern [120]	s_f	C_{max}			×			Johnson	t_j
Wang and Cheng [121]	s_f	T_{max}				×		constr.	mwt
Bagchi et al. [122]	s_{fg}	C_{max}							modeled as TSP
Logendran et al. [60]	s_{fg}	C_{max}					×	TS	new lower bound
Logendran and Sriskandarajah [123]	s_{fg}	$\sum C_j$					×	TS	carryover sequence-dependency
Logendran et al. [124]	s_{fg}	$\sum C_j$					×	TS	carryover sequence-dependency
Liou and Liu [125]	s_{fg}	$\sum C_j$					×	PSO	t_j, new encoding scheme
Liou et al. [58]	s_{fg}	$\sum C_j$					×	hybrid	t_j, new encoding scheme
Sabouni and Logendran [97]	s_{fg}	$\sum w_jT_j, \sum w_jF_j$	×	×			×	GA, TS, other	carryover sequence-dependency
Sabouni and Logendran [126]	s_{fg}	$\sum w_jT_j, \sum w_jF_j$	×		×		×	B&B, TS	r_j
Sabouni and Logendran [127]	s_{fg}	$\sum w_jT_j, \sum w_jF_j$	×				×	GA	carryover sequence-dependency

s_{fg}: sequence-dependent setups; s_f: sequence-independent setups; MIP: mixed integer programming model; Johnson: modified Johnson's rule; B&B: branch and bound; constr.: constructive algorithm; TS: tabu search; PSO: particle swarm optimization; hybrid: hybrid metaheuristic; GA: genetic algorithm; other: other heuristic.

polynomial time algorithm. Logendran [34] gave an overview of several published constructive heuristics for the $F2|s_f, prmu, gta|C_{max}$ problem.

In general, nearly all publications on the two-machine group scheduling problem with sequence-independent setup times focused on the minimization of makespan. An exception is the work by Wang and Cheng [121] who minimized maximum lateness and considered a no-wait restriction, solving $F2|s_f, prmu, gta, nwt|L_{max}$ problem. For this problem several properties and dominance relations between jobs are identified and a constructive heuristic is presented.

Two-machine flowshop group scheduling problems with sequence-dependent setup times were considered explicitly only after 2003 as before that they were part of the general m-machine case. Logendran et al. [124] were the first to observe a practical case from PCB manufacturing with setups of the special case of carryover sequence-dependency. They minimized mean flow time with the help of a taboo search algorithm which is based on a taboo tree. The results were compared to a rather weak lower bound. Sabouni and Logendran [97] took up this issue and introduced an additional external setup time for the kitting operation in the assembly of PCBs, so that two types of setup times were integrated. A mixed-integer programming model was presented which minimizes a bi-criteria objective function of weighted tardiness and weighted flow time. In order to solve larger problems a very time-efficient novel Forward Improving Exchanges/Inserts algorithm (FIEI) was developed and compared to a genetic algorithm as well as a taboo search algorithm. The same problem was extended by adding dynamic job release times to propose a taboo search algorithm with a branching decision tree algorithm to evaluate the integrated sequences [126]. A genetic algorithm was also applied to the dynamic assembly system and tested for several test instances by Sabouni and Logendran [127].

Bagchi et al. [122] presented the work by Logendran and Sriskandarajah [128], who modeled the schedules of each family as blocks and, building on that, transformed the problem into a traveling salesman problem. Based on this and the results from Logendran and Sriskandarajah [123], Logendran et al. [60] developed three taboo search metaheuristics to solve $F2|s_{fg}, prmu, gta|C_{max}$, which were compared to each other. The use of a long-term memory led to superior results, while a mathematical programming based lower bounding mechanism could reduce the computational effort significantly. The block structure of the families' schedules was used for the development of a new lower bound. Liou and Liu [125] proposed a particle swarm optimization algorithm which was extended to a hybrid algorithm with a genetic algorithm by Liou et al. [58] in order to minimize the sum of completion times with removal and job transportation times integrated. The algorithm is based on a new encoding scheme, which uses a single job sequence containing all jobs regardless of their specific part family as well as a group vector, that is necessary for decoding a solution. Furthermore, they developed three lower bounds for this problem.

5.2. The m-machine flowshop

5.2.1. Sequence-independent setup times

Makespan minimization: In the beginning, research on the group scheduling problem with multiple machines focused on sequence-independent setup times and makespan minimization, $Fm|s_f, prmu, gta|C_{max}$. Hitomi and Ham [23] were the first to apply a modification of Petrov's PT algorithm [129] to this problem. Furthermore, Ham et al. [2] presented a new lower bound, a branch and bound algorithm as well as several theorems and dominance conditions that provided a basis for further group scheduling literature. In order to generate improved group schedules Allison [130] was the first to combine two constructive heuristics originally designed to solve general flowshop problems, one for each level of scheduling. Subsequently, this has become the basis for constructive heuristics solving the m-machine group scheduling problem and has resulted in several studies identifying the improved combination of algorithms. Allison [130] compared all four possible combinations of PT and CDS algorithm [95]. The use of the CDS algorithm for solving both levels showed superior results. Logendran et al. [131] extended this analysis by including the constructive heuristic LN by Logendran and Nudtasomboon [71]. In their study, the combination algorithm LN for sequencing jobs and algorithm PT for sequencing part families outperformed other algorithms. Furthermore, it was ascertained that on the first level multiple-pass heuristics, i.e. heuristics such as CDS or LN, that generate several solutions per iteration from which the best is chosen, generally tend to be superior compared to single-pass heuristics, which generate a single solution per iteration only. A summary of several constructive heuristics can be found in Logendran [34].

The first metaheuristic approach was developed by Vakharia and Chang [132], who proposed a simulated annealing algorithm which outperformed the known extensions of NEH and CDS algorithms. Furthermore, it was shown that the branch and bound algorithm by Ham et al. [2] could only optimally solve small problem instances with up to 5 families, 5 jobs per family and 5 machines within reasonable CPU time. The simulated annealing algorithm was compared to a novel taboo search algorithm by Skorin-Kapov and Vakharia [133]. While for small instances similar results were gained, the taboo search algorithm led to significantly lower makespans for larger problems requiring less computational effort. Sridhar and Rajendran [134] considered missing operations in flowshop environments explicitly and introduced a modified recursive equation for calculating makespan. Their genetic algorithm, that is also used to minimize total flow time, led to superior results compared to the simulated annealing algorithm by Vakharia and Chang [132].

A comprehensive review of 15 constructive heuristics as well as metaheuristic approaches for $Fm|s_f, prmu, gta|C_{max}$ was performed by Schaller [10]. In addition to the adjusted procedures of CDS and NEH, a local search algorithm, the taboo search heuristic by Skorin-Kapov and Vakharia [133], the genetic algorithm by Sridhar and Rajendran [134] as well as several variations and combinations of these algorithms were compared. Furthermore, a new tight lower bound was developed requiring less computational time. In this study an adjusted taboo search led to best results whereas constructive heuristics were highly efficient in terms of computational effort. A lower bound was developed by Schaller [135]. For several test instances it was proven to be superior to previous lower bounds if used within a branch and bound algorithm adding a local search algorithm for determining job sequences. These results were extended by Schaller [136] to develop an efficient branch and bound algorithm. Three branch and bound-based heuristics were introduced, whereas the so-called HOA heuristic was shown to outperform the previously best known taboo search algorithm. After that, only Solimanpur and Elmi [137] considered the case of sequence-independent setup times with makespan criterion. The proposed mixed-integer programming model and their taboo search heuristic takes limited buffers into account. The results for several test instances were compared to SVS algorithm which was originally developed for the cell scheduling problem (see Section 7).

Other criteria: Except for the genetic algorithm by Sridhar and Rajendran [134], which is used for minimizing makespan as well as total flow time, only few researches have addressed other criteria for flowshop group scheduling problems with sequence-

Table 4
Multiple machine flowshop group scheduling literature with sequence-independent setups.

Publication	Objective	Bi-/multi-criteria	MIP	Exact algorithm	Constr. algorithm	Meta-heuristic	Solution method	Comment
Hitomi and Ham [23]	C_{max}			×	×		B&B, constr.	
Ham et al. [2]	C_{max}			×	×		B&B, Johnson, constr.	
Allison [130]	C_{max}				×		constr.	
Vakharia and Chang [132]	C_{max}					×	SA	
Skorin-Kapov and Vakharia [133]	C_{max}					×	TS	
Logendran et al. [131]	C_{max}				×		constr.	
Logendran [34]	C_{max}				×		constr.	
Schaller [10]	C_{max}				×		constr.	
Schaller [135]	C_{max}			×			B&B, B&B-heuristic	new lower bound
Schaller [136]	C_{max}			×			B&B, B&B-heuristic	
Solimanpur and Elmi [137]	C_{max}		×			×	TS	*block*
Zolfaghari and Liang [138]	$\sum w_j T_j$		×			×	TS, SA, hybrid	machine speed selection included
Gupta and Schaller [139]	$\sum C_j$			×	×	×	B&B, constr., GA, TS, other	
Sridhar and Rajendran [134]	C_{max}, $\sum C_j$	×				×	GA	missing operations considered

MIP: mixed integer programming model; Johnson: modified Johnson's rule; B&B: branch and bound; B&B-heuristic: branch and bound based heuristic; constr.: constructive algorithm; SA: simulated annealing; TS: tabu search; hybrid: hybrid metaheuristic; GA: genetic algorithm; other: other heuristic.

Table 5
Multiple machine flowshop group scheduling literature with sequence-dependent setups – I.

Publication	Objective	Bi-/multi-criteria	MIP	Exact algorithm	Constr. algorithm	Meta-heuristic	Solution method	Comment
Schaller et al. [18]	C_{max}				×		constr.	new lower bound
Das and Canel [145]	C_{max}		×	×			B&B	multi-cell FMS solved as GS
Franca et al. [53]	C_{max}					×	GA, MA, other	
Salmasi and Logendran [146]	C_{max}					×	TS	
Eddaly et al. [37]	C_{max}					×	ILS, EDA	
Hendizadeh et al. [36]	C_{max}					×	TS	
Lin et al. [143]	C_{max}					×	SA	
Celano et al. [14]	C_{max}				×	×	constr., GA, TS	*block*
Bouabda et al. [101]	C_{max}					×	GA	
Bouabda et al. [147]	C_{max}					×	ILS	
Cheng and Ying [38]	C_{max}					×	ILS	
Lin et al. [39]	C_{max}					×	SA	
Neufeld [148]	C_{max}					×	VNS	splitting of families allowed
Salmasi et al. [99]	C_{max}		×			×	ACO	
Ying et al. [149]	C_{max}		×			×	GA, SA, IG	*nwt*
Costa et al. [150]	C_{max}		×			×	GA	worker assignment included
Eddaly et al. [142]	C_{max}					×	EDA, ILS	
Chen et al. [56]	C_{max}					×	ACO, GA	new problem decoding
Chen et al. [57]	C_{max}					×	TS	new problem decoding
Costa et al. [151]	C_{max}		×			×	GA	worker assignment included
Li and Li [152]	C_{max}					×	HHS	
Balaji and Porselvi [153]	C_{max}					×	SA, AIS	multi-cell FMS solved as GS
Nikjo and Rezaeian [154]	C_{max}		×			×	SA	r_j
Nikjo and Zarook [55]	C_{max}					×	GA, TS	non-permutation schedules, r_j
Neufeld et al. [141]	C_{max}				×		constr.	
Fichera et al. [155]	C_{max}		×			×	GA	setups with learning effects
Liou and Hsieh [59]	C_{max}					×	hybrid	t_j

MIP: mixed integer programming model; constr.: constructive algorithm; B&B: branch and bound; GA: genetic algorithm; MA: memetic algorithm; ILS: iterated local search; TS: tabu search; SA: simulated annealing; VNS: variable neighborhood search; ACO: ant colony optimization; AIS: artificial immune system; IG: iterated greedy algorithm; EDA: estimation of distribution algorithm; PSO: particle swarm optimization; CG: column generation; HHS: hybrid harmony search; hybrid: hybrid metaheuristic; other: other metaheuristic.

independent setup times. Zolfaghari and Liang [138] studied a group scheduling problem with machine speed selection included minimizing weighted tardiness. A non-linear mixed-integer programming model as well as an iterative hybrid algorithm using taboo search and simulated annealing was developed.

Gupta and Schaller [139] presented a branch and bound algorithm as well as several constructive and metaheuristic approaches for the $Fm|s_g, prmu, gta|\sum C_j$ problem. A comparison to the genetic algorithm by Sridhar and Rajendran [134] proves the efficiency of the newly proposed algorithms.

Table 6
Multiple machine flowshop group scheduling literature with sequence-dependent setups – II.

Publication	Objective	Bi-/multi-criteria	MIP	Exact algorithm	Constr. algorithm	Meta-heuristic	Solution method	Comment
Cho and Ahn [156]	$\sum T_j$					×	hybrid	
Lin et al. [54]	$C_{max}, T_{max}, \sum T_j, \sum w_jT_j, \sum C_j, \sum w_jC_j$					×	SA, TS, GA	non-permutation schedules
Ying et al. [144]	$C_{max}, T_{max}, \sum T_j, \sum w_jT_j, \sum C_j, \sum w_jC_j$					×	SA	non-permutation schedules
Gelogullari and Logendran [20]	$\sum C_j$		×			×	TS, CG	carryover sequence-dependency
Salmasi et al. [11]	$\sum C_j$		×	×		×	B&P, ACO, TS	new lower bound
Hajinejad et al. [157]	$\sum C_j$					×	PSO, hybrid	
Naderi and Salmasi [98]	$\sum C_j$		×			×	hybrid	
Villadiego et al. [158]	$\sum C_j$					×	ILS	
Costa et al. [159]	$\sum C_j$					×	hybrid	
Ibrahem et al. [160]	$\sum C_j$					×	PSO, GA	robust optimization
Keshavarz and Salmasi [161]	$\sum C_j$					×	hybrid, B&P	new lower bound
Hendizadeh et al. [162]	C_{max}, C_j	×				×	GA	new lower bound
Taghavifard et al. [163]	C_{max}, $\sum T_j$	×	×			×	GA	learning effects
Lin and Ying [164]	C_{max}, $\sum C_j$, $\sum T_j$	×				×	SA	
Lu and Logendran [96]	$\sum w_jT_j$, $\sum w_jF_j$	×	×			×	TS	r_j, *unavail*
Li et al. [1]	$\sum T_j$, $\sum F_j$	×				×	HHS	

MIP: mixed integer programming model; constr.: constructive algorithm; B&P: branch and price; GA: genetic algorithm; MA: memetic algorithm; ILS: iterated local search; TS: tabu search; SA: simulated annealing; VNS: variable neighborhood search; ACO: ant colony optimization; IG: iterated greedy algorithm; EDA: estimation of distribution algorithm; PSO: particle swarm optimization; CG: column generation; HHS: hybrid harmony search; hybrid: hybrid metaheuristic; other: other metaheuristic.

All publications on group scheduling problems with multiple machines and sequence-independent setup times are listed in Table 4.

5.2.2. *Sequence-dependent setup times*

Makespan minimization: While sequence-independent setup times have been the starting point for group scheduling research, recent publications have focused on sequence-dependent setups almost entirely. A summary of these is given in Tables 5 and 6. Again, most emphasis was put on makespan minimization, which is considered in various algorithms to solve the $Fm|s_{fg}, prmu, gta|C_{max}$ problem. Analogous to the study on sequence-independent setups, Schaller et al. [18] provided a comprehensive evaluation of 12 constructive two-stage heuristics based on the algorithms NEH and CMD as well as those by Gupta and Darrow [43], Gupta [140] and Baker [119]. Moreover, a further improvement of the generated solutions was achieved by a simple local search algorithm. The development of lower bounds and test instances enabled an evaluation of the proposed solution methods. Among these, the so-called CMD algorithm performed best, which applies CDS algorithm for determining job sequences, while the family sequence is identified by a modified NEH algorithm. Finally, the local search improvement algorithm is applied. Since then, the proposed test problems with up to 10 machines, 10 families and 10 jobs per family have widely been used as benchmark instances for evaluating new algorithms. Recently, Neufeld et al. [141] pointed out some specific characteristics of the group scheduling problem based on the structure of a families schedule and the role of inserted idle times. Using these properties in NEH-based algorithms leads to significant improvements compared to CMD.

While no other development of constructive heuristics was conducted, the application and comparison of metaheuristic approaches became the focus of attention. Franca et al. [53] developed a multi-start algorithm and evolutionary algorithms that use a structured population. A memetic algorithm which includes a local search procedure showed the best performance to minimize makespan. Similar results were obtained by applying other metaheuristics. Hendizadeh et al. [36] proposed five variations of taboo search algorithms with the concept of elitism and the acceptance of inferior solutions as well as sophisticated diversification and intensification methods. Eddaly et al. [37] introduced an estimation of distribution algorithm that is followed by an iterated local search. Variations of this algorithm were also presented by Eddaly et al. [142]. A simulated annealing algorithm by Lin et al. [143], that uses a Cauchy function instead of Boltzmann function during the annealing process, showed to be very efficient. Lin et al. [54] adjusted this approach in order to generate non-permutation schedules within the family sequence, but not within the job sequence of each family. Ying et al. [144] provided an extended comparison and analysis of the benefits of non-permutation schedules. Besides makespan, for which little improvements could only be obtained, other optimization criteria were also taken into account. Furthermore, a combined multi-start hill climbing strategy and simulated annealing procedure was published by Lin et al. [39]. Their proposed algorithm produced similar results compared to eight existing metaheuristics. The idea of non-permutation schedules is extended to the job sequences by Nikjo and Zarook [55]. Besides a MIP formulation, a genetic algorithm as well as a taboo search heuristic is presented and compared for own test instances of the $Fm|s_{fg}, gta, r_j|C_{max}$ problem. For a relaxed version of the problem the algorithms are compared to the iterated greedy algorithm of Cheng and Ying [38], however cannot lead to superior results. Apart from these papers no other research considering the general $Fm|s_{fg}, gta|\gamma$ including non-permutation schedules have been published.

Bouabda et al. [101] enhanced a genetic algorithm with a branch and bound procedure which, again, led to group schedules with slightly lower makespans. In addition, another likewise efficient two-level iterated greedy heuristic was proposed by Cheng and Ying [38] and a nested iterated local search by Bouabda et al. [147]. Neufeld [148] introduced a variable neighborhood search algorithm which allowed part families to be split in separated batches and analyzed the use of non-exhaustive algorithms. According to his results, exhaustive schedules could not be improved efficiently by a split processing of part families. However, considering relative performance compared to a lower bound, the VNS algorithms were not as effective as previous metaheuristic approaches. Besides, a hybrid harmony search algorithm presented by Li and Li [152] solved the group scheduling problem with sequence-dependent setup times as effectively as the best-performing previous algorithms, even though requiring additional computational effort.

For the benchmark instances by Schaller et al. [18], the average deviations from a common lower bound for all state-of-the-art metaheuristics are always less than 3% and in total at about 0.5–0.8% within reasonable CPU time. This proves the effectiveness of these approaches. However, the computational efficiency of the studied algorithms is not always comparable, as different coding languages and hardware configurations have been used.

Moreover, several other algorithms have been developed that have not been compared to existing metaheuristics using benchmark problem instances or using the authors' own problem instances only. Among these is a taboo search approach with an initial solution generator by Salmasi and Logendran [146]. Based on ant colony optimization and NEH, Salmasi et al. [99] developed an efficient hybrid metaheuristic. For alleged realistic problems with 2, 3 or 6 machines, comparable results to the memetic algorithm by Franca et al. [53] were obtained. For the first time a mathematical model is presented to solve the $Fm|s_{fg}, gta, prmu|C_{max}$ problem. Additionally, a novel lower bounding method is proposed. Nikjo and Rezaeian [154] propose a non-linear mathematical model based on the work by Stafford and Tseng [165] and Ying et al. [149] that integrates varying release times of jobs. Furthermore, a simulated annealing algorithm is used to solve own test instances. By the use of ant colony optimization and genetic algorithm by Chen et al. [56] as well as taboo search heuristic by Chen et al. [57] a novel decoding scheme is analyzed. While the results for solving the benchmark instances from Schaller et al. [18] indicate that the proposed solution representation can be used effectively, the algorithms have not been compared to existing lower bounds or state of the art metaheuristics yet. Liou and Hsieh [59] proposed a hybrid metaheuristic, integrating elements of genetic algorithms as well as particle swarm optimization, for solving the $Fm|s_{fg}, gta, prmu, t_j|C_{max}$ problem. Based on the developments of Logendran et al. [102], three adjusted lower bounds were presented for evaluating their approach. Das and Canel [145] introduced a mixed-integer programming model and a branch and bound algorithm for scheduling a multi-cell flexible manufacturing system. Nevertheless, as each cell is modelled as a machine in a flowshop, this problem equals a group scheduling problem. The proposed branch and bound algorithm was evaluated by the required number of nodes using own problem instances. The same problem was studied by Balaji and Porselvi [153]. They introduce a simulated annealing as well as artificial immune system algorithm.

Extended versions of the flowshop group scheduling problem were considered by other authors as well. Celano et al. [14] integrated a limited buffer capacity and analyzed the effect of buffer capacity on solution quality. For their own problem instances of $Fm|s_{fg}, prmu, gta, block|C_{max}$, their genetic algorithm outperformed a modified NEH algorithm as well as a taboo search heuristic. Ying et al. [149] added a no-wait constraint to solve $Fm|s_{fg}, prmu, gta, nwt|C_{max}$. Besides a MIP formulation, several metaheuristic algorithms were presented. Among these, a simulated annealing showed best results. Additionally, Coast et al. [150,151] integrated the worker assignment problem to the scheduling task in manufacturing cells. They proposed a mixed integer linear programming model as well as genetic algorithms, for which extended test instances from Salmasi et al. [11] were compared to a hybrid particle swarm optimization algorithm originally developed to minimize total flow time by Hajinejad et al. [157]. Fichera et al. [155] present a hybrid genetic algorithm for a group scheduling problem with learning effects on setup times. 270 instances similar to those presented by Salmasi et al. [11] are generated, whose setup times decrease depending on the order of processing part families. For these test instances the novel algorithm outperforms the hybrid particle swarm optimization approach by Salmasi et al. [99] as well as the simulated annealing algorithm by Naderi and Salmasi [98].

Other criteria: Besides makespan minimization, other optimization criteria came to the center of attention recently. Cho and Ahn [156] introduced a hybrid algorithm in order to solve $Fm|s_{fg}, prmu, gta|\sum T_j$. While all job sequences are determined with priority rules, a genetic algorithm is used for sequencing part families. As mentioned before, the simulated annealing algorithm by Lin et al. [54] generated non-permutation schedules considering several optimization criteria, which are maximum lateness, total tardiness, weighted tardiness, total flow time, weighted flow time and makespan. While for flow time based criteria, non-permutation schedules led to about 1% lower objective functions compared to permutation schedules only, the average improvement for due date based criteria was about 10%.

Total flow time or total completion time were also considered in recent publications. Salmasi et al. [11] were the first to take $Fm|s_{fg}, prmu, gta|\sum C_j$ into account by applying a hybrid ant colony optimization heuristic similar to Salmasi et al. [99] as well as a taboo search algorithm. The effectiveness of the developed algorithms is compared to a new lower bound, that is determined by a novel lower bounding mechanism based on branch and price technique. In doing so, the good quality of the gained upper bounds could be proven. Moreover, a first mathematical model for this problem is proposed. While Hajinejad et al. [157] proposed a hybrid PSO algorithm that outperformed the ACO algorithm by Salmasi et al. [11], further improvements could be obtained by Villadiego et al. [158], who presented an iterated local search, which adapts the concept of reference solutions from scatter search. Comparable results were produced by a hybrid algorithm using genetic algorithm and simulated annealing by Naderi and Salmasi [98], who also proposed two MIP formulations for total completion time criterion. By one of these mathematical models a remarkable improvement of efficiency can be gained compared to existing formulations. This is proven by solving significantly larger problem instances with up to 60 jobs optimally. For benchmark instances from Salmasi et al. [11], another hybrid algorithm based on genetic algorithm, that was enhanced by ideas from Biased Random Sampling search technique, could lead to significantly better results compared to previous approaches [159]. Ibrahem et al. [160] also minimized total flow time with particle swarm optimization as well as genetic algorithm and analyzed particularly the algorithms' robustness using Design of Experiments technique. A new state-of-the-art lower bounding method based on branch and price as well as a hybrid genetic algorithm was presented by Keshavarz and Salmasi [161] solving the permutation flowshop group scheduling problem with minimization of total completion time. The proposed lower bound as well as the hybrid algorithm is proven to have a better performance than previous methods from the literature. Gelogullari and Logendran [20] developed a general framework for the group scheduling problem with carryover sequence-dependent setup times for various optimization criteria and presented a MIP formulation as well as a comprehensive analysis of several taboo search based metaheuristics minimizing total flow time. Furthermore, a column generation approach was applied. All algorithms were evaluated by a new lower bound for several setup configurations.

Bi-criteria optimization: As a strong need for optimizing several criteria arises in operational practice, bi- or multi-criteria objective functions have also been applied to group scheduling algorithms. Hendizadeh et al. [162] developed a multi-objective genetic algorithm in order to minimize makespan and total flow time simultaneously. By modifying the lower bounds developed for sequence-independent setups and total flow time criteria by Gupta and Schaller [139], the algorithm is evaluated using benchmark instances by Schaller et al. [18]. For the same test problems, a two-level multi-start simulated annealing algorithm by Lin et al. [164] could gain superior results concerning several performance measures. Furthermore, the same algorithm was applied to solve the $Fm|s_{fg}, prmu, gta|\sum T_j, C_{max}$ problem.

Taghavifard et al. [163] minimized the sum of makespan and total tardiness, too, by a MIP formulation as well as genetic algorithms. For random instances with learning effects integrated, a non-dominated sorting and a non-dominated rank genetic algorithm were compared. Weighted tardiness as well as weighted completion time with job release dates and constrained machine availability was considered by Lu and Logendran [96]. They introduced a MIP formulation and compared the results of ten heuristic algorithms. Among these a taboo search algorithm with two initial solution finding mechanisms showed the best performance. Li et al. [1] proposed an efficient hybrid harmony search algorithm minimizing total weighted flow time as well as total tardiness. This new algorithm outperformed other adapted metaheuristic approaches significantly.

6. Static group scheduling in flexible flowshops

The existence of parallel machines at some stages of manufacturing cells results in the flexible flowshop group scheduling problem. Even though parallel machines are quite common in practical manufacturing environments, they had not been studied theoretically in group scheduling literature for a long time. Hence, the first publications arose from practical cases of a blade production line in an airplane engine plant by Li [15] and Huang and Li [166], furniture manufacturing by Wilson et al [12] as well as tile industry by Andres et al. [167]. The studied problems mainly show very specific characteristics derived from the respective real world application. Only in 2005 Logendran et al. [45] were the first to analyze constructive heuristics for a flexible flowshop with sequence-independent family setup times on a more general, theoretical level. Table 7 lists all publications on group scheduling in flexible flowshops.

Sequence-independent setups: Group scheduling with sequence-independent setup times was considered by very few researchers only, all minimizing makespan. Huang and Li [166] were the first to discuss a special case of a two-stage flexible flowshop with a single machine on the first stage, while the second stage consists of up to eight uniform, parallel machines ($FF2, (0, Qm^2)|s_f, gta|C_{max}$). For this problem Huang and Li developed a lower bound and applied two modified simple heuristics based on dispatching rules, as originally presented by Li [15]. A Forward Heuristic, that initially sequences all jobs at the first stage and, afterwards, assigns jobs to machines at the second stage, leads to superior results compared to a Backward Heuristic, that assigns jobs to machines on the second stage before sequencing them on the first machine. Wilson et al. [12] study a flexible group scheduling flowshop with two stages and two identical machines at each stage and analyze the influence of a split processing of jobs from the same family on each parallel machine, which leads to multiple setups for the respective part family. For this, variations of genetic algorithms are proposed using different splitting and sequencing methods. The results indicate that lower makespans can be gained by effectively splitting part families even though the number of additional setups per family is usually less than two. Logendran et al. [45] presented a fundamental work providing a basis for future publications for group scheduling in flexible flowshops. They proposed three combinations of the constructive heuristics PT (algorithm from Petrov [129]) and LN (Logendran and Nudtasomboon [71]), partly allowing multiple setups on parallel machines for jobs from the same part family. Comparing these algorithms to each other, it was shown that schedules with multiple setups can lead to significantly superior results especially for large size problems. Furthermore, different influencing factors are statistically evaluated. The structure of the generated new test instances formed a basis for several following studies on flexible flowshop group scheduling.

Sequence-dependent setups: Apart from these publications solely sequence-dependent setup times have been regarded. Based on the same practical application presented by Huang and Li [166], Li [15] developed simple heuristics for a $FF2, (0, Pm^2)$ flexible flowshop as well as a lower bounding method. Additionally, minor

Table 7
Static group scheduling in flexible flowshops literature.

Publication	Setup type	Stages	Objective	Bi-/multi-criteria	MIP	Constr. algorithm	Meta-heuristic	Solution method	Comment
Huang and Li [166]	s_f	2	C_{max}			×		other	uniform machines, new LB
Wilson et al. [12]	s_f	2	C_{max}				×	GA	multiple setups
Logendran et al. [45]	s_f	m	C_{max}			×		constr.	multiple setups
Li [15]	s_{fg}	2	C_{max}			×		other	multiple setups, new LB
Adressi et al. [168]	s_{fg}	2	C_{max}				×	SA, GA	*nwt*, *unavail*
Andres [167]	s_{fg}	3	$\sum C_j$			×		constr.	
Logendran et al. [102]	s_{fg}	m	C_{max}				×	TS	
Zandieh et al. [65]	s_{fg}	m	C_{max}				×	SA, GA	missing operations considered
Karimi et al. [66]	s_{fg}	m	C_{max}				×	ICA	missing operations considered
Shahvari et al. [61]	s_{fg}	m	C_{max}		×		×	TS	
Keshavarz and Salmasi [62]	s_{fg}	m	C_{max}		×		×	MA	new LB
Khamseh et al. [68]	s_{fg}	m	C_{max}				×	SA, GA	*unavail*
Keshavarz et al. [63]	s_{fg}	m	$\sum C_j$		×		×	MA	
Behnamian et al. [46]	s_{fg}	m	$\sum T_j$		×		×	hybrid	missing operations considered
Karimi et al. [47]	s_{fg}	m	C_{max}, $\sum w_jT_j$	×	×		×	GA	missing operations considered
Zandieh and Karimi [169]	s_{fg}	m	C_{max}, $\sum w_jT_j$	×	×		×	GA	missing operations considered
Fadaei and Zandieh [170]	s_{fg}	m	C_{max}, $\sum T_j$	×			×	GA	
Ebrahimi et al. [67]	s_{fg}	m	C_{max}, $\sum T_j$	×			×	GA	uncertain due dates
Bozorgirad and Logendran [69]	s_{fg}	m	$\sum w_jT_j$, $\sum w_jC_j$	×	×		×	TS	uniform machines
Bozorgirad and Logendran [19]	s_{fg}	m	$\sum w_jT_j$, $\sum w_jF_j$	×	×		×	LS, GA	unrelated machines, NPS, r_j, *unavail*, M_j, learning effects

MIP: mixed integer programming model; constr.: constructive algorithm; hybrid: hybrid metaheuristic; NPS: non-permutation schedules; LB: lower bound; GA: genetic algorithm; MA: memetic algorithm; TS: tabu search; SA: simulated annealing; ICA: imperialist competitive algorithm; other: other heuristic.

setups for every changeover of jobs from the same family as well as multiple setups were taken into account. Besides, Li discussed the case, when parts cannot be transferred to the second stage individually unless a whole lot is completed at the first stage. Including the no-wait restriction as well as random machine breakdowns, Adressi et al. [168] solve the $FF2,(Pm^1,Pm^2)|s_{fg},gta, nwt, unavail|C_{max}$ problem by a simulated annealing algorithm and a genetic algorithm, which are compared to each other. Andres et al. [167] present a case study of a $FF3,(P4^1,P2^2,P3^3)|s_{fg},gta|C_{max}$ problem. The main focus is put on the grouping of parts to families in order to reduce setup times, nonetheless, subsequently a simple heuristic based on dispatching rule is presented for sequencing jobs and part families. The effectiveness of the algorithm is analyzed concerning flow time and the resulting setup times.

Apart from these two case studies most research was conducted on the $FFc,(Pm^l)_{l=1}^c)|s_{fg},gta|C_{max}$ problem. Logendran et al. [102] triggered future publications by developing three metaheuristic algorithms based on tabu search to solve this problem for the first time. Following the original idea of group technology and avoiding setup times a single setup were only allowed, i.e. at each stage all parts from a family were assigned to the same machine. In the proposed algorithms all generated sequences are definite on the first machines only. On all following machines FIFO rule is applied, which can lead to non-permutation schedules. Based on the test instances by Logendran et al. [45], that were adjusted to sequence-dependent setup times, several factors for the heuristics' performance were identified: while problem size and the systems flexibility show a significant influence and interaction, the initial solution was proven not to be significant for the gained makespans and computation times. Eventually, several other metaheuristic approaches have been proposed. Zandieh et al. [65] presented two novel metaheuristics that show superior results compared to the taboo search approaches by Logendran et al. [102]. While a genetic algorithm based metaheuristic generated schedules with least makespan, the simulated annealing algorithm was proven to be very time-efficient. The test instances used by Zandieh et al. [65] explicitly consider that jobs may skip some stages. Shavari et al. [61] presented a linear mathematical model as well as six new metaheuristics based on tabu search. The best performing algorithm gained similar results compared to the existing metaheuristics by Logendran et al. [102] for small size problems, whereas a statistically significant improvement was proven for medium and large size problems. Again non-permutation schedules become possible due to the use of FIFO rule on subsequent stages. A memetic algorithm by Keshavarz and Salmasi [62] outperformed the metaheuristic by Shavari et al. using 45 test instances similar to [102]. Besides, Keshavarz and Salmasi developed mixed integer linear mathematical model and a first lower bound for the $FFc,(Pm^l)_{l=1}^c)|s_{fg},gta|C_{max}$ problem. Compared to this lower bound the memetic algorithm showed 0.8% relative percentage deviation for small size problems and about 5% for medium sized problems with up to 65 jobs. Other metaheuristic approaches, that have not been compared to the existing algorithms mentioned above, were published by Karimi et al. [66]. An imperialist competitive algorithm, for which Taguchi method was used for finding an optimal parameter setting, as well as an adjusted random key genetic algorithm was tested, whereby the imperialist competitive algorithm showed best results. Khamseh et al. [68] integrate preventive maintenance activities, that lead to machine unavailability, to the same problem. For this, a simulated annealing metaheuristic and a more effective genetic algorithm are proposed.

Besides makespan, for flexible flowshops with identical machines other criteria have been considered rarely only. Behnamian et al. [46] presented a hybrid metaheuristic for minimizing the sum of the earliness and tardiness of jobs taking due windows into account. The hybrid metaheuristic integrates elements of particle swarm optimization, simulated annealing as well as variable neighborhood search and was compared to the algorithms by Zandieh et al. [65], that had been developed for makespan criterion. Keshavarz et al. [63] applied a memetic algorithm (MA), similar to Keshavarz and Salmasi [62], to solve the $FFc,(Pm^l)_{l=1}^c)|s_{fg},gta|\sum C_j$ problem for the first time. Furthermore, a linear model and a branch and price based lower bounding method were proposed. Compared to this lower bound the memetic algorithm gained solutions with an average percentage deviation of about 6%.

Recently, bi-criteria optimization problems received increased attention. Karimi et al. [47] studied the flexible flowshop group scheduling problem with identical machines and a simultaneous minimization of makespan and total weighted tardiness. Again, the possibility that jobs may skip some stages was considered. Two multi-phase genetic algorithms were developed using Pareto archive concepts. By design of experiment method a suitable parameter setting was gained to solve the problem more effectively compared to conventional multi-objective genetic algorithms. Moreover, a mathematical formulation is presented to solve small problems. For the same problem Zandieh et al. [169] developed a new multi-population genetic algorithm that outperformed the metaheuristics presented by Karimi et al. [47]. Variations of genetic algorithms regarding a bi-criteria objective function with makespan and total tardiness were presented by Ebrahimi et al. [67] and Fadaei et al. [170]. Ebrahimi et al. [67] assumed that due dates are uncertain and follow a normal distribution. A non-dominated sorting genetic algorithm as well as multi-objective genetic algorithm was compared to the multi-phase genetic algorithm by Karimi et al. [47] and especially the first showed superior performance. Similarly, out of three different population based algorithms a non-dominated sorting genetic algorithm by Fadaei et al. [170] led to best solutions, however, existing algorithms from the literature had not been considered in this comparative study.

To the best of our knowledge only three publications by Bozorgirad and Logendran consider flexible flowshop group scheduling with sequence-dependent setup times and non-identical parallel machines, all minimizing a bi-criteria objective function with total weighted tardiness and total weighted flowtime. For the case of uniform parallel machines a MIP formulation and three taboo search algorithms were compared by Bozorgirad and Logendran [69]. For small problems the metaheuristics showed mainly less than 3% deviation from the optimal solution and were proven to be very time efficient even for large size problems. Bozorgirad and Logendran [171] also proposed a tight lower bound based on column generation for flexible flowshop group scheduling problems with unrelated parallel machines. Besides, various local search and genetic algorithm based approaches were presented for this problem, adding release times, machine eligibility restrictions, machine ready times as well as position-based processing time with learning effect [19]. Two local search algorithms were designed to generate non-permutation schedules, however, showed a comparably weak performance. The results indicated that genetic algorithms are more suitable for solving this problem.

In general, apart from Huang and Li [166] and Bozorgirad and Logendran [69] no studies with uniform machines were published, while Bozorgirad and Logendran [19,171] are the only studies on unrelated machines. So far, mainly population based metaheuristics have been applied to solve flexible flowshop group scheduling problems. Constructive algorithms or optimizing algorithms have been applied rarely. Furthermore, besides makespan no other criterion has been studied extensively. Nevertheless, even for makespan the gained average deviations from lower bounds for flexible flowshop group scheduling problems remain rather

Table 8
Multiple-cell scheduling literature.

Publication	Setup type	Objective	Bi-/multi-criteria	MIP	Exact algorithm	Constr. algorithm	Meta-heuristic	Solution method	Comment
Solimanpur et al. [49]	s_f	C_{max}				×		constr.	
Saravanan and Noorul Haq [176]	s_f	C_{max}					×	SS	
Tavakkoli-Moghaddam et al. [177]	s_f	EE, C_{max}	×	×			×	GA, MA	t_j, family assignment integrated
Saravanan and Noorul Haq [178]	s_f	C_{max}, $\sum C_j$					×	SS	
Mosbah and Dao [179]	s_f	C_{max}, $\sum C_j$	×				×	hybrid	
Solimanpur and Elmi [103]	s_f	C_{max}		×			×	TS	
Tavakkoli-Moghaddam et al. [180]	s_{fg}	*cost*	×	×			×	SS	family assignment integrated
Taghavifard [181]	s_{fg}	C_{max}					×	ACO, hybrid	

MIP: mixed integer programming model; *EE*: exceptional elements (parts with intercellular moves); constr.: constructive algorithm; SS: scatter search; GA: genetic algorithm; MA: memetic algorithm; TS: tabu search; ACO: ant colony optimization; hybrid: hybrid metaheuristic; other: other metaheuristic.

large compared to those realized for the basic $Fm|s_{fg}, gta, prmu|C_{max}$ problem with usually less than 0.8%.

7. Static group scheduling in multiple cells

As mentioned in Section 2.2 the scheduling task in cellular manufacturing systems with multiple cells and intercellular movement is generally modeled similar to flowshop group scheduling and is referred to as cell scheduling problem. Despite the strong similarities to classical group scheduling approaches, the cell scheduling problem exhibits some distinct characteristics that justify the development of problem specific algorithms. Hence, algorithms for scheduling multiple cells have been developed, solving job shop cells [172–174], single stage cells with reentrant material flows [175] as well as flowshop cells. All publications on the cell scheduling problem are summarized in Table 8.

Sequence-independent setups: Solimanpur et al. [49] were the first to define the multiple-cell scheduling problem with sequence-independent setup times and makespan criterion. After providing a problem description, they introduced a constructive algorithm, referred to as SVS, which solves the intra-cell scheduling by NEH and inter-cell scheduling by a pairwise comparison procedure. The SVS algorithm was compared to the group scheduling algorithm LN-PT, which was outperformed for several test instances from the literature. Based on this work, Saravanan and Noorul Haq [176,178] proposed the solution of the first level of scheduling by a scatter search algorithm, which led to better results for three out of seven test instances.

A nested tabu search algorithm for the multiple-cell scheduling problem with sequence-independent setups was developed by Solimanpur and Elmi [103]. The constructive SVS algorithm was outperformed for test problems from Solimanpur et al. [49]. Besides, a mixed-integer linear programming model was introduced to solve small instances optimally.

Mosbah and Dao [179] present an extended great deluge approach, which was used for minimizing makespan as well as a multi-criteria objective function, considering makespan, flow time and tardiness. A multi-objective optimization function was also used by Tavakkoli-Moghaddam et al. [177] for jointly solving the multiple-cell scheduling problem integrating the assignment of parts to cells. For this problem, a MIP formulation as well as genetic and memetic algorithm was introduced.

Sequence-dependent setups: Multiple-cell scheduling with sequence-dependent setup times has been considered only recently by Tavakkoli-Moghaddam et al. [180], who, again, jointly solved the assignment as well as scheduling problem in flowline manufacturing cells. A nonlinear mathematical programming model and a scatter search algorithm were used to minimize a bi-criteria cost function. Similar approaches were presented by Gholipour-Kanani et al. [182,183]. For makespan minimization Taghavifard [181] applied several genetic, ant colony optimization and local search based algorithms. However, the best-performing ant colony optimization algorithm led to slightly better schedules compared to the SVS algorithm only. To the best of our knowledge, no further approaches have been published to solve the cell scheduling problem with sequence-dependent setup times.

8. Conclusions and future research

This review gives a comprehensive overview of the available literature on flowshop group scheduling problems, that are typically found in CMS. It is classified on the basis of the nature of the shop environment (dynamic or static environment), the number of parallel machines at each stage (flowshop or flexible flowshop), the considered number of independent manufacturing cells (single or multiple cells) as well as machines (two or more-than-two machines) and the kind of setup times (sequence-independent or sequence-dependent setups). While simulation studies in dynamic environments formed the basis for the following research in static environments, various constructive and metaheuristic algorithms have proven to be suitable to solve problems related to group scheduling.

Due to its wide range of application especially in manufacturing industry, there is a still growing interest on group scheduling research. Hence, our results can point to some fruitful directions for further studies. First, there is still a need for considerations of different optimization criteria. While effective metaheuristic for the classical flowshop group scheduling problem with sequence-dependent and sequence-independent setup times minimizing makespan have been applied already, especially tardiness and earliness related criteria have been regarded by few researchers only, despite their high relevance in manufacturing industry. Furthermore, problems with multi-criteria objective functions, that meet the requirements of operational practice more appropriately, should be taken into account for detailed problem analyses and the development of specific algorithms. Particularly for flexible flowshop group scheduling problems besides makespan different objective functions were studied occasionally only, so that no conclusive evaluation of existing solution approaches is possible so far. Other criteria, such as total weighted completion time or maximum lateness, have not been considered at all.

Non-permutation schedules can enhance the shop floor performance. So far, non-permutation sequences have been applied for the family sequence only. Especially, since missing operations are common in CM, differing job sequences on each machine give reason to expect promising improvements for various criteria. Different streams of research, especially in flowshops with sequence-dependent setups, should be brought together and the proposed solution approaches have to be compared on a comparable basis.

Furthermore, nearly no exact algorithms, such as branch and bound, as well as constructive heuristics have been applied solving group scheduling problems in flexible flowshops. With most proposed metaheuristics based on genetic algorithms, the use of different metaheuristic approaches might be worthwhile.

Besides, especially for flexible flowshop group scheduling and cell scheduling problems there is still a need for commonly established benchmark instances. Furthermore, for existing comparisons of algorithms rarely statistical methods such as *t*-test or analysis of variance (ANOVA) have been applied. This could provide a solid basis influencing factors as well as a specific evaluation of developed algorithms. Apart from that, the use of different solution representation might still provide room for designing more efficient (meta-)heuristic approaches. Besides, the mathematical model proposed by Naderi and Salmasi [98], that is able to solve significantly larger test instances, indicates that the development of more efficient formulation is worthwhile and promising.

Moreover, several constraints that have partly been integrated in classical scheduling problems already should be applied to group scheduling as well, such as no-wait constraints, limited buffers, learning effects or the consideration of non-anticipatory setups. This would ensure the applicability of the developed algorithms. Integrated approaches for jointly solving group scheduling, cell formation and cell design have been proposed recently. However, none of these approaches take specific characteristics of part families, such as family setup times, into account. By combining these problems some additional improvements may be gained. Similar to research on classical flexible flowshop problem, in group scheduling uniform and unrelated parallel machines have been studied rarely only. Also, sequence-independent family setup times have been taken into account rarely in environments with parallel machines.

As inter-cellular movement is common in practical CMS, the multiple-cell scheduling problem should receive additional attention. So far the assignment of only a single family per cell has been regarded and problem characteristics resulting from the autonomy of the cells have not been pointed out. When considering multiple cells, non-permutation schedules concerning the cell sequence seem to be reasonable as different cells are intended to be organized independently. Besides, most publications assume a unidirectional material flow even for machines from different cells, which is not the case in many manufacturing environments.

Finally, due to the fast growing number of publications on scheduling and related topics, further problem-specific reviews could help to keep track of the developments in research. Among these should be reviews of literature on scheduling flexible manufacturing systems, flowshop group scheduling of identical jobs (also known as batch scheduling of identical jobs) and the class of lot-streaming problems, group scheduling with single machines as well as cell layout in CM. Thus, group scheduling research provides an excellent opportunity for making many more contributions and to improve the practical application of the developed techniques.

References

[1] Li Y, Li X, Gupta J. Solving the multi-objective flowline manufacturing cell scheduling problem by hybrid harmony search. Expert Syst Appl 2015;42(3):1409–17. http://dx.doi.org/10.1016/j.eswa.2014.09.007.
[2] Ham I, Hitomi K, Yoshida T. Group technologyapplications to production management. Dordrecht, Boston, London: Kluwer Academic Publishing; 1985.
[3] Wu T, Chung S, Chang C. Hybrid simulated annealing algorithm with mutation operator to the cell formation problem with alternative process routings. Expert Syst Appl 2009;36(2P2):3652–61.
[4] Snead C. Group technology. New York: Van Nostrand Reinhold; 1989.
[5] Askin R, Iyer A. A comparison of scheduling philosophies for manufacturing cells. Eur J Oper Res 1993;69(3):438–49.
[6] Wu X, Chu C-H, Wang Y, Yue D. Genetic algorithms for integrating cell formation with machine layout and scheduling. Comput Ind Eng 2007;53(2):277–89.
[7] Akturk M. Joint cell loading and scheduling approach to cellular manufacturing systems. Int J Prod Res 2011;49(21):6321–41.
[8] Papaioannou G, Wilson J. The evolution of cell formation problem methodologies based on recent studies (1997–2008): review and directions for future research. Eur J Oper Res 2010;206(3):509–21. http://dx.doi.org/10.1016/j.ejor.2009.10.020.
[9] Kia R, Baboli A, Javadian N, Tavakkoli-Moghaddam R, Kazemi M, Khorrami J. Solving a group layout design model of a dynamic cellular manufacturing system with alternative process routings, lot splitting and flexible reconfiguration by simulated annealing. Comput Oper Res 2012;39(11):2642–58.
[10] Schaller J. A comparison of heuristics for family and job scheduling in a flowline manufacturing cell. Int J Prod Res 2000;38(2):287–308.
[11] Salmasi N, Logendran R, Skandari M. Total flow time minimization in a flowshop sequence-dependent group scheduling problem. Comput Oper Res 2010;37(1):199–212.
[12] Wilson A, King R, Hodgson T. Scheduling non-similar groups on a flow line: multiple group setups. Robot Comput Integr Manuf 2004;20(6):505–15.
[13] Lin H-T, Liao C-J. A case study in a two-stage hybrid flow shop with setup time and dedicated machines. Int J Prod Econ 2003;86(2):133–43.
[14] Celano G, Costa A, Fichera S. Constrained scheduling of the inspection activities on semiconductor wafers grouped in families with sequence-dependent set-up times. Int J Adv Manuf Technol 2010;46(5–8):695–705.
[15] Li S. A hybrid two-stage flowshop with part family, batch production, major and minor set-ups. Eur J Oper Res 1997;102(1):142–56.
[16] van der Zee D. Family based dispatching with batch availability. Int J Prod Res 2013;51(12):3643–53.
[17] van der Zee D-J, Gaalman G, Nomden G. Family based dispatching in manufacturing networks. Int J Prod Res 2011;49(23):7059–84.
[18] Schaller J, Gupta J, Vakharia A. Scheduling a flowline manufacturing cell with sequence dependent family setup times. Eur J Oper Res 2000;125(2):324–39.
[19] Bozorgirad M, Logendran R. A comparison of local search algorithms with population-based algorithms in hybrid flow shop scheduling problems with realistic characteristics. Int J Adv Manuf Technol 2015:1–17.
[20] Gelogullari C, Logendran R. Group-scheduling problems in electronics manufacturing. J Sched 2010;13(2):177–202.
[21] Yang D, Kuo W, Chern M. Multi-family scheduling in a two-machine reentrant flow shop with setups. Eur J Oper Res 2008;187(3):1160–70.
[22] MacCawley A. The international wine supply chain: challenges from bottling to the glass [Ph.D. thesis]. Georgia Institute of Technology; 2014.
[23] Hitomi K, Ham I. Operations scheduling for group technology applications. Ann CIRP 1976;25(1):419–22.
[24] Vaithianathan R, McRoberts KL. On scheduling in a gt environment. J Manuf Syst 1982;1(2):149–55.
[25] Mahmoodi F, Dooley K. A comparison of exhaustive and non-exhaustive group scheduling heuristics in a manufacturing cell. Int J Prod Res 1991;29(9):1923–39.
[26] Flynn B. Repetitive lots: the use of a sequence-dependent set-up time scheduling procedure in group technology and traditional shops. J Oper Manag 1987;7(1–2):203–16.
[27] Allahverdi A, Gupta J, Aldowaisan T. A review of scheduling research involving setup considerations. Omega 1999;27(2):219–39.
[28] Allahverdi A, Ng C, Cheng T, Kovalyov M. A survey of scheduling problems with setup times or costs. Eur J Oper Res 2008;187(3):985–1032.
[29] Allahverdi A. The third comprehensive survey on scheduling problems with setup times/costs. Eur J Oper Res 2015;246(2):345–78.
[30] Cheng T, Gupta J, Wang G. A review of flowshop scheduling research with setup times. Prod Oper Manag 2000;9(3):262–82.
[31] Zhu X, Wilhelm W. Scheduling and lot sizing with sequence-dependent setup: a literature review. IIE Trans 2006;38(11):987–1007.
[32] Filho M, Morais M, Boiko T, Miyata H, Varolo F. Scheduling in flow shop with sequence-dependent setup times: literature review and analysis. Int J Bus Innov Res 2013;7(4):466–86.
[33] Potts C, Kovalyov M. Scheduling with batching: a review. Eur J Oper Res 2000;120(2):228–49.
[34] Logendran R. Group scheduling for cellular manufacturing systems. In: Group technology and cellular manufacturing, Methodologies and applications. New York: Gordon and Breach Science Publishers; 1998. p. 153–208.
[35] Rajendran C. A heuristic for scheduling in flowshop and flowline-based manufacturing cell with multi-criteria. Int J Prod Res 1994;32(11):2541–58.
[36] Hendizadeh S, Faramarzi H, Mansouri S, Gupta J, El Mekkawy T. Meta-heuristics for scheduling a flowline manufacturing cell with sequence dependent family setup times. Int J Prod Econ 2008;111(2):593–605.
[37] Eddaly M, Jarboui B, Bouabda R, Rebai A. Hybrid estimation of distribution algorithm for permutation flowshop scheduling problem with sequence dependent family setup times. In: International conference on computers & industrial engineering, 2009, CIE 2009. Troyes: IEEE; 2009. p. 217–20.

[38] Cheng H-M, Ying K-C. Minimizing makespan in a flow-line manufacturing cell with sequence dependent family setup times. Expert Syst Appl 2011;38(12):15517–22.
[39] Lin S-W, Ying K-C, Lu C-C, Gupta J. Applying multi-start simulated annealing to schedule a flowline manufacturing cell with sequence dependent family setup times. Int J Prod Econ 2011;130(2):246–54.
[40] Garey M, Johnson D, Sethi R. The complexity of flowshop and jobshop scheduling. Math Oper Res 1976;1(2):117–29.
[41] Lenstra J, Rinnooy Kan A, Brucker P. Complexity of machine scheduling problems. Ann Discrete Math 1977;1:343–62.
[42] Koulamas C. The total tardiness problem: review and extensions. Oper Res 1994;42(6):1025–41.
[43] Gupta J, Darrow W. The two-machine sequence dependent flowshop scheduling problem. Eur J Oper Res 1986;24(3):439–46.
[44] Kleinau U. Two-machine shop scheduling problems with batch processing. Math Comput Model 1993;17(6):55–66.
[45] Logendran R, Carson S, Hanson E. Group scheduling in flexible flow shops. Int J Prod Econ 2005;96(2):143–55.
[46] Behnamian J, Zandieh M, Fatemi Ghomi S. Due windows group scheduling using an effective hybrid optimization approach. Int J Adv Manuf Technol 2010;46(5–8):721–35.
[47] Karimi N, Zandieh M, Karamooz H. Bi-objective group scheduling in hybrid flexible flowshop: a multi-phase approach. Expert Syst Appl 2010;37(6):4024–32.
[48] Wang S, Sarker B. Locating cells with bottleneck machines in cellular manufacturing systems. Int J Prod Res 2002;40(2):403–24.
[49] Solimanpur M, Vrat P, Shankar R. A heuristic to minimize makespan of cell scheduling problem. Int J Prod Econ 2004;88(3):231–41.
[50] Graham R, Lawler E, Lenstra J, Kan A. Optimization and approximation in deterministic sequencing and scheduling: a survey. Ann Discrete Math 1979;5(2):287–326.
[51] Pinedo M. Scheduling—theory, algorithms and systems, 4th edition, vol. 2. New Jersey: Springer; 2012.
[52] Ribas I, Leisten R, Framinan J. Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. Comput Oper Res 2010;37(8):1439–54.
[53] Franca P, Gupta J, Mendes A, Moscato P, Veltink K. Evolutionary algorithms for scheduling a flowshop manufacturing cell with sequence dependent family setups. Comput Ind Eng 2005;48(3):491–506.
[54] Lin S-W, Ying K-C, Lee Z. Metaheuristics for scheduling a non-permutation flowline manufacturing cell with sequence dependent family setup times. Comput Oper Res 2009;36(4):1110–21.
[55] Nikjo B, Zarook Y. A non-permutation flow shop manufacturing cell scheduling problem with part's sequence dependent family setup times. Int J Appl Metaheuristic Comput 2014;5(4):70–86.
[56] Chen C-F, Wu M-C, Lin K-H. Effect of solution representations on tabu search in scheduling applications,. Comput Oper Res 2013;40(12):2817–25. http://dx.doi.org/10.1016/j.cor.2013.06.003.
[57] Chen C-F, Wu M-C, Li Y-H, Tai P-H, Chiou C-W. A comparison of two chromosome representation schemes used in solving a family-based scheduling problem. Robot Comput-Integr Manuf 2013;29(3):21–30.
[58] Liou C-D, Hsieh Y-C, Chen Y-Y. A new encoding scheme-based hybrid algorithm for minimising two-machine flow-shop group scheduling problem. Int J Syst Sci 2013;44(1):77–93.
[59] Liou C-D, Hsieh Y-C. A hybrid algorithm for the multi-stage flow shop group scheduling with sequence-dependent setup and transportation times. Int J Prod Econ 2015;170(Part A):258–67.
[60] Logendran R, Salmasi N, Sriskandarajah C. Two-machine group scheduling problems in discrete 1020 parts manufacturing with sequence-dependent setups. Comput Oper Res 2006;33(1):158–80.
[61] Shahvari O, Salmasi N, Logendran R, Abbasi B. An efficient tabu search algorithm for flexible flow shop sequence-dependent group scheduling problems. Int J Prod Res 2012;50(15):4237–54.
[62] Keshavarz T, Salmasi N. Makespan minimisation in flexible flowshop sequence-dependent group scheduling problem. Int J Prod Res 2013;51(20):6182–93.
[63] Keshavarz T, Salmasi N, Varmazyar M. Minimizing total completion time in the flexible flowshop sequence-dependent group scheduling problem. Ann Oper Res 2015;226(1):351–77.
[64] Norman B, Bean J. A genetic algorithm methodology for complex scheduling problems. Naval Res Logist 1999;46(2):199–211.
[65] Zandieh M, Dorri B, Khamseh A. Robust metaheuristics for group scheduling with sequence-dependent setup times in hybrid flexible flow shops. Int J Adv Manuf Technol 2009;43(7–8):767–78.
[66] Karimi N, Zandieh M, Najafi A. Group scheduling in flexible flow shops: a hybridised approach of imperialist competitive algorithm and electromagnetic-like mechanism. Int J Prod Res 2011;49(16):4965–77.
[67] Ebrahimi M, Fatemi Ghomi S, Karimi B. Hybrid flow shop scheduling with sequence dependent family setup time and uncertain due dates. Appl Math Model 2014;38(9):2490–504.
[68] Khamseh A, Jolai F, Babaei M. Integrating sequence-dependent group scheduling problem and preventive maintenance in flexible flow shops. Int J Adv Manuf Technol 2015;77(1–4):173–85.
[69] Bozorgirad M, Logendran R. Bi-criteria group scheduling in hybrid flowshops. Int J Prod Econ 2013;145(2):599–612.
[70] Radharamanan R. A heuristic algorithm for group scheduling. Comput Ind Eng 1986;11(1):204–8.
[71] Logendran R, Nudtasomboon N. Minimizing the makespan of a group scheduling problem: a new heuristic. Int J Prod Econ 1991;22(3):217–30.
[72] Logendran R. Group scheduling problem: key to flexible manufacturing systems. Comput Ind Eng 1992;23(1):113–6.
[73] Sridhar J, Rajendran C. Scheduling in flowshop and cellular manufacturing systems with multiple objectives a genetic algorithmic approach. Prod Plan Control 1996;7(4):374–82.
[74] Rajendran C, Ziegler H. Heuristics for scheduling in flowshops and flowline-based manufacturing cells to minimize the sum of weighted flowtime and weighted tardiness of jobs. Comput Ind Eng 1999;37(4):671–90.
[75] Venkataramanaiah S. Scheduling in cellular manufacturing systems: an heuristic approach. Int J Prod Res 2008;46(2):429–49.
[76] Henneberg M, Neufeld J. A constructive algorithm and a simulated annealing approach for solving flowshop problems with missing operations. Int J Prod Res 2015:1–17. http://dx.doi.org/10.1080/00207543.2015.1082670.
[77] Cetinkaya F, Kayaligil M. Unit sized transfer batch scheduling with setup times. Comput Ind Eng 1992;22(2):177–83.
[78] Logendran R, Sriskandarajah C. Two-machine group scheduling problem with blocking and anticipatory setups. Eur J Oper Res 1993;69(3):467–81.
[79] Greene T, Sadowski R. Mixed integer program for loading and scheduling multiple flexible manufacturing cells. Eur J Oper Res 1986;24(3):379–86.
[80] Vickson R, Alfredsson B. Two-and three-machine flow shop scheduling problems with equal sized transfer batches. Int J Prod Res 1992;30(7):1551–74.
[81] Sotskov Y, Tautenhahn T, Werner F. Heuristics for permutation flow shop scheduling with batch setup times. OR Spektr 1996;18(2):67–80.
[82] Danneberg D, Tautenhahn T, Werner F. A comparison of heuristic algorithms for flow shop scheduling problems with setup times and limited batch size. Math Comput Model 1999;29(9):101–26.
[83] Wang X, Cheng T. Two-machine flowshop scheduling with job class setups to minimize total flowtime. Comput Oper Res 2005;32(11):2751–70.
[84] Wang M-C, Rao Y-Q, Wang K-P. A niche genetic algorithm for two-machine flowshop scheduling with family sequence-dependent setup times and a common due window. In: 2010 IEEE international conference on industrial engineering and engineering management (IEEM). Singapore: IEEE; 2010. p. 296–300.
[85] Schaller J. Scheduling a permutation flow shop with family setups to minimise total tardiness. Int J Prod Res 2012;50(8):2204–17.
[86] Pang K-W. A genetic algorithm based heuristic for two machine no-wait flowshop scheduling problems with class setup times that minimizes maximum lateness. Int J Prod Econ 2013;141(1):127–36.
[87] Shen L, Gupta J, Buscher U. Flow shop batching and scheduling with sequence-dependent setup times. J Sched 2014;17(4):353–70. http://dx.doi.org/10.1007/s10951-014-0369-x.
[88] Yang W. Group scheduling in a two-stage flowshop. J Oper Res Soc 2002;53(12):1367–73.
[89] Nekoiemehr N, Selvarajah E, Zhang G. Scheduling of jobs with cross families in two stage manufacturing systems. Int J Prod Econ 2015;167:88–96.
[90] Pranzo M. Batch scheduling in a two-machine flow shop with limited buffer and sequence independent setup times and removal times. Eur J Oper Res 2004;153(3):581–92.
[91] Pan Q-P, Ruiz R. An estimation of distribution algorithm for lot-streaming flow shop problems with setup times. Omega 2012;40(2):166–80.
[92] Mohammadi M, Fatemi Ghomi S. Genetic algorithm-based heuristic for capacitated lotsizing problem in flow shops with sequence-dependent setups. Expert Syst Appl 2011;38(6):7201–7.
[93] Wemmerlöv U, Vakharia A. Job and family scheduling of a flow-line manufacturing cell: a simulation study. IIE Trans 1991;23(4):383–93.
[94] Nawaz M, Enscore EE, Ham I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. Omega 1983;11(1):91–5.
[95] Campbell H, Smith M, Dudek R. A heuristic algorithm for n jobs, m machine sequencing problem. Manag Sci 1970;16(10):13630–7.
[96] Lu D, Logendran R. Bi-criteria group scheduling with sequence-dependent setup time in a flow shop. J Oper Res Soc 2013;64(4):530–46.
[97] Sabouni M, Logendran R. Carryover sequence-dependent group scheduling with the integration of internal and external setup times. Eur J Oper Res 2013;224(1):8–22.
[98] Naderi B, Salmasi N. Permutation flowshops in group scheduling with sequence-dependent setup times. Eur J Ind Eng 2012;6(2):177–98.
[99] Salmasi N, Logendran R, Skandari M. Makespan minimization of a flowshop sequence-dependent group scheduling problem. Int J Adv Manuf Technol 2011;56(5–8):699–710.
[100] Johnson S. Optimal two- and three-stage production schedules with setup times included. Naval Res Logist 1954;1:61–8.
[101] Bouabda R, Jarboui B, Eddaly M, Rebaï A. A branch and bound enhanced genetic algorithm for scheduling a flowline manufacturing cell with sequence dependent family setup times. Comput Oper Res 2011;38(1):387–93. http://dx.doi.org/10.1016/j.cor.2010.06.006.
[102] Logendran R, deSzoeke P, Barnard F. Sequence-dependent group scheduling problems in flexible flow shops. Int J Prod Econ 2006;102(1):66–86.
[103] Solimanpur M, Elmi A. A tabu search approach for cell scheduling problem with makespan criterion. Int J Prod Econ 2013;141:639–45.
[104] Mosier C, Elvers D, Kelly D. Analysis of group technology scheduling heuristics. Int J Prod Res 1984;22(5):857–75.

[105] Flynn B. The effects of setup time on output capacity in cellular manufacturing. Int J Prod Res 1987;25(12):1761–72.
[106] Mahmoodi F, Dooley K, Starr P. An investigation of dynamic group scheduling heuristics in a job shop manufacturing cell. Int J Prod Res 1990;28(9):1695–711.
[107] Mahmoodi F, Dooley K, Starr P. An evaluation of order releasing and due date assignment heuristics in a cellular manufacturing system. J Oper Manag 1990;9(4):548–73.
[108] Ruben R, Mosier C, Mahmoodi F. A comprehensive analysis of group scheduling heuristics in a job shop cell. Int J Prod Res 1993;31(6):1343–69.
[109] Wirth G, Mahmoodi F, Mosier C. An investigation of scheduling policies in a dual-constrained manufacturing cell. Decis Sci 1993;24(4):761–88.
[110] Kannan V, Lyman S. Impact of family-based scheduling on transfer batches in a job shop manufacturing cell. Int J Prod Res 1994;32(12):2777–94.
[111] Ponnambalam S, Aravindan P, Reddy K. Analysis of group-scheduling heuristics in a manufacturing cell. Int J Adv Manuf Technol 1999;15(12):914–32.
[112] Hitomi K, Nakamura N, Yoshida T, Okuda K. An experimental investigation of group production scheduling. In: Proceedings of the fourth international conference on production research; 1977. p. 608–17.
[113] Russell G, Philipoom P. Sequencing rules and due date setting procedures in flow line cells with family setups. J Oper Manag 1990;10(4):524–45.
[114] Mahmoodi F, Tierney EJ, Mosier C. Dynamic group scheduling heuristics in a flow-through cell environment. Decis Sci 1992;23(1):61–85.
[115] Frazier G. An evaluation of group scheduling heuristics in a flow-line manufacturing cell. Int J Prod Res 1996;34(4):959–76.
[116] Reddy V, Narendran T. Heuristics and sequence-dependent set-up jobs in flow line cells. Int J Prod Res 2003;41(1):193–206.
[117] Yoshida T, Hitomi K. Optimal two-stage production scheduling with setup times separated. IIE Trans 1979;11(3):261–3.
[118] Sekiguchi Y. Optimal schedule in a gt-type flow-shop under series-parallel precedence constraints. J Oper Res Soc Jpn 1983;26(3):226–52.
[119] Baker K. Scheduling groups of jobs in the two-machine flow shop. Math Comput Model 1990;13(3):29–36.
[120] Yang D, Chern M. Two-machine flowshop group scheduling problem. Comput Oper Res 2000;27(10):975–85.
[121] Wang X, Cheng T. A heuristic approach for tow-machine no-wait flowshop scheduling with due dates and class setups. Comput Oper Res 2006;33(5):1326–44.
[122] Bagchi T, Gupta J, Sriskandarajah C. A review of tsp based approaches for flowshop scheduling. Eur J Oper Res 2006;169(3):816–54.
[123] Logendran R, Sriskandarajah C. Minimizing the total completion time in a two-machine group scheduling problem with carryover sequence dependency; 2001.
[124] Logendran R, Gelogullari C, Sriskandarajah C. Minimizing the mean flow time in a two-machine group-scheduling problem with carryover sequence dependency. Robot Comput Integr Manuf 2003;19(1–2):21–33.
[125] Liou C-D, Liu C-H. A novel encoding scheme of PSO for two-machine group scheduling In: Advances in swarm intelligence. Berlin: Springer; 2010. p. 128–34.
[126] Sabouni M, Logendran R. PCB assembly in a multi-machine flowshop with dynamic board release times. Advances in sustainable and competitive manufacturing systems. Basel: Springer; 2013. p. 1063–76.
[127] Sabouni M, Logendran R. Genetic algorithm application in a dynamic PCB assembly with carryover sequence-dependent setups. Int J Mech Aerosp Ind Mechatron Manuf Eng 2013;7(4):740–8.
[128] Logendran R, Sriskandarajah C. Scheduling methodology for hardware manufacturing systems. In: Proceedings of the NSF design, service, and manufacturing grantees and research conference, Birmingham, AL; 2003.
[129] Petrov V. Flowline group production planning. London: Business Publications; 1968.
[130] Allison J. Combining Petrov's heuristic and the cds heuristic in group scheduling problems. Comput Ind Eng 1990;19(1–4):457–61.
[131] Logendran R, Mai L, Talkington D. Combined heuristics for bi-level group scheduling problems. Int J Prod Econ 1995;38(2–3):133–45.
[132] Vakharia A, Chang Y. A simulated annealing approach to scheduling a manufacturing cell. Naval Res Logist 1990;37(4):559–77.
[133] Skorin-Kapov J, Vakharia A. Scheduling a flow-line manufacturing cell: a tabu search approach. Int J Prod Res 1993;31(7):1721–34.
[134] Sridhar J, Rajendran C. A genetic algorithm for family and job scheduling in a flowline-based manufacturing cell. Comput Ind Eng 1994;27:469–72.
[135] Schaller J. A new lower bound for the flow shop group scheduling problem. Comput Ind Eng 2001;41(2):151–61.
[136] Schaller J. An improved branch and bound procedure for scheduling a flow line manufacturing cell. Int J Prod Res 2005;43(22):4697–720.
[137] Solimanpur M, Elmi A. A tabu search approach for group scheduling in buffer-constrained flow shop cells. Int J Comput Integr Manuf 2011;24(3):257–68.
[138] Zolfaghari S, Liang M. Jointly solving the group scheduling and machining speed selection problems: a hybrid tabu search and simulated annealing approach. Int J Prod Res 1999;37(10):2377–97.
[139] Gupta J, Schaller J. Minimizing flow time in a flow-line manufacturing cell with family setup times. J Oper Res Soc 2006;57(2):163–76.
[140] Gupta J. Flowshop schedules with sequence dependent setup times. J Oper Res Soc Jpn 1986;29(3):206–19.
[141] Neufeld J, Gupta J, Buscher U. Minimizing makespan in flowshop group scheduling with sequence-dependent family setup times using inserted idle times. Int J Prod Res 2015;53(6):1791–806. http://dx.doi.org/10.1080/00207543.2014.961209.
[142] Eddaly M, Jarboui B, Bouabda R, Siarry P, Rebaï A. An estimation of distribution algorithm for solving flow shop scheduling problems with sequence-dependent family setup times. In: Metaheuristics for production scheduling, vol. 1. Wiley Online Library, Hoboken; 2013. p. 1–17.
[143] Lin S-W, Gupta J, Ying K-C, Lee Z. Using simulated annealing to schedule a flowshop manufacturing cell with sequence-dependent family setup times. Int J Prod Res 2009;47(12):3205–17.
[144] Ying K-C, Gupta J, Lin S-W, Lee Z-L. Permutation and non-permutation schedules for the flowline manufacturing cell with sequence dependent family setups. Int J Prod Res 2010;48(8):2169–84.
[145] Das S, Canel C. An algorithm for scheduling batches of parts in a multi-cell flexible manufacturing system. Int J Prod Econ 2005;97(3):247–62.
[146] Salmasi N, Logendran R. A heuristic approach for multi-stage sequence-dependent group scheduling problems. J Ind Eng Int Islam Azad Univ 2008;4(7):48–58.
[147] Bouabda R, Jarboui B, Rebaï A. A nested iterated local search algorithm for scheduling a flowline manufacturing cell with sequence dependent family setup times. In: 2011 Fourth international conference on logistics (LOGISTIQUA). Hammamet: IEEE; 2011. p. 526–31.
[148] Neufeld J. Group scheduling in flow-line manufacturing cells with variable neighborhood search. In: Dynamics and sustainability in international logistics and supply chain management: proceedings of the sixth German-Russian logistics and SCM workshop DR-LOG 2011 in Bremen; 2011. p. 274–84.
[149] Ying K-C, Lee Z-L, Lu C-C, Lin S-W. Metaheuristics for scheduling a no-wait flowshop manufacturing cell with sequence-dependent family setups. Int J Adv Manuf Technol 2012;58(5–8):671–82.
[150] Costa A, Fichera S, Cappadonna F. A genetic algorithm for scheduling both job families and skilled workforce. Int J Oper Quant Manag 2013;19(4):221–47.
[151] Costa A, Cappadonna F, Fichera S. Joint optimization of a flow-shop group scheduling with sequence dependent set-up times and skilled workforce assignment. Int J Prod Res 2014;52(9):2696–728.
[152] Li Y, Li X. Hybrid harmony search for the flowline manufacturing cell scheduling problem. In: Proceedings of the 2014 IEEE 18th international conference on computer supported cooperative work in design (CSCWD). Hsinchu: IEEE; 2014. p. 214–9.
[153] Balaji A, Porselvi S. Artificial immune system algorithm and simulated annealing algorithm for scheduling batches of parts based on job availability model in a multi-cell flexible manufacturing system. Proc Eng 2014;97:1524–33.
[154] Nikjo B, Rezaeian J. Meta heuristic for minimizing makespan in a flow-line manufacturing cell with sequence dependent family setup times. J Optim Ind Eng 2014;7(16):21–9.
[155] Fichera S, Costa A, Cappadonna F. Scheduling jobs families with learning effect on the setup. Adv Oper Res 2015;2015:1–12.
[156] Cho K, Ahn B. A hybrid genetic algorithm for group scheduling with sequence dependent group setup time. Int J Ind Eng 2003;10:442–8.
[157] Hajinejad D, Salmasi N, Mokhtari R. A fast hybrid particle swarm optimization algorithm for flow shop sequence dependent group scheduling problem. Sci Iran 2011;18(3):759–64.
[158] Villadiego H, Arroyo J, Jacob V, dos Santos A, Goncalves L. An efficient ils heuristic for total flow time minimization in a flow shop sequence dependent group scheduling problem. In: 2012 12th International conference on hybrid intelligent systems (HIS). Pune: IEEE; 2012. p. 259–64.
[159] Costa A, Cappadonna F, Fichera S. A hybrid metaheuristic approach for minimizing the total flow time in a flow shop sequence dependent group scheduling problem. Algorithms 2014;7(3):376–96.
[160] Ibrahem A, Elmekkawy T, Peng Q. Robust metaheuristics for scheduling cellular flowshop with family sequence-dependent setup times. Proc CIRP 2014;17:428–33.
[161] Keshavarz T, Salmasi N. Efficient upper and lower bounding methods for flowshop sequence-dependent group scheduling problems. Eur J Ind Eng 2014;8(3):366–87.
[162] Hendizadeh S, El Mekkawy T, Wang G. Bi-criteria scheduling of a flowshop manufacturing cell with sequence dependent setup times. Eur J Ind Eng 2007;1(4):391–413.
[163] Taghavifard M, Javanshir H, Roueintan M, Soleimany E. Multi-objective group scheduling with learning effect in the cellular manufacturing system. Int J Ind Eng Comput 2011;2(3):617–30.
[164] Lin S-W, Ying K-C. Scheduling a bi-criteria flowshop manufacturing cell with sequence-dependent family setup times. Eur J Ind Eng 2012;6(4):474–96.
[165] Stafford E, Tseng F. Two models for a family of flowshop sequencing problems. Eur J Oper Res 2002;142(2):282–93.
[166] Huang W, Li S. A two-stage hybrid flowshop with uniform machines and setup times. Math Comput Model 1998;27(2):27–45.
[167] Andres C, Albarracin J, Tormo G, Vicens E, Garcia-Sabater J. Group technology in a hybrid flowshop environment: a case study. Eur J Oper Res 2005;167(1):272–81.
[168] Adressi A, Hassanpour S, Azizi V. Solving group scheduling problem in no-wait flexible flowshop with random machine breakdown. Decis Sci Lett 2016;5(1):157–68.
[169] Zandieh M, Karimi N. An adaptive multi-population genetic algorithm to solve the multi-objective group scheduling problem in hybrid flexible flowshop with sequence-dependent setup times. J Intell Manuf 2011;22(6):979–89.

[170] Fadaei M, Zandieh M. Scheduling a bi-objective hybrid flow shop with sequence-dependent family setup times using metaheuristics. Arab J Sci Eng 2013;38(8):2233–44.
[171] Bozorgirad M, Logendran R. Developing tight lower bounds for hybrid flow shop scheduling problems. In: IIE annual conference proceedings. Montreal: Institute of industrial engineers; 2014. p. 341.
[172] Tang J, Wang X, Kaku I, Yung K-L. Optimization of parts scheduling in multiple cells considering intercell move using scatter search approach. J Intell Manuf 2010;21:525–37.
[173] Elmi A, Solimanpur M, Topaloglu S, Elmi A. A simulated annealing algorithm for the job shop cell scheduling problem with intercellular moves and reentrant parts. Comput Ind Eng 2011;61(1):171–8.
[174] Li D, Wang Y, Xiao G, Tang J. Dynamic parts scheduling in multiple job shop cells considering intercell moves and flexible routes. Comput Oper Res 2013;40(5):1207–23.
[175] Yang W, Liao C. Group scheduling on two cells with intercell movement. Comput Oper Res 1996;23(10):997–1006.
[176] Saravanan M, Haq ANoorul. A scatter search method to minimise makespan of cell scheduling problem. Int J Agile Syst Manag 2008;3(1):18–36.
[177] Tavakkoli-Moghaddam R, Gholipour-Kanani Y, Cheraghalizadeh R. A genetic algorithm and memetic algorithm to sequencing and scheduling of cellular manufacturing systems. Int J Manag Sci Eng Manag 2008;3(2):119–30.
[178] Saravanan M, Noorul Haq A. Comparison of metaheuristic methods to minimize makespan of cell scheduling problems. Int J Des Manuf Technol 2009;3(2):77–82.
[179] Mosbah A, Dao T-M. Optimimization of group scheduling using simulation with the meta-heuristic extended great deluge (EGD) approach. In: 2010 IEEE international conference on industrial engineering and engineering management (IEEM). Singapore: IEEE; 2010. p. 275–80.
[180] Tavakkoli-Moghaddam R, Javadian N, Khorrami A, Gholipour-Kanani Y. Design of a scatter search method for a novel multi-criteria group scheduling problem in a cellular manufacturing system. Expert Syst Appl 2010;37(3):2661–9.
[181] Taghavifard M. Scheduling cellular manufacturing systems using aco and ga. Int J Appl Metaheuristic Comput 2012;3(1):48–64.
[182] Gholipour-Kanani Y, Tavakkoli-Moghaddam R, Khorrami A. Solving a multi-criteria group scheduling problem for a cellular manufacturing system by scatter search. J Chin Inst Ind Eng 2011;28(3):192–205.
[183] Gholipour-Kanani Y, Tavakkoli-Moghaddam R, Cheraghalizadeh R, Babol I, Mahmoodjanloo M. A new mathematical model for a multi-criteria group scheduling problem in a cms solved by a branch-and-bound method. In: Proceedings of the 2012 international conference on industrial engineering and operations management; 2012.