PERGAMON

Omega, Int. J. Mgmt Sci. 27 (1999) 219–239

omega
The International Journal of Management Science

A review of scheduling research involving setup considerations

Ali Allahverdi[a,*], Jatinder N.D. Gupta[b,1], Tariq Aldowaisan[a,2]

[a]Department of Mechanical and Industrial Engineering, Kuwait University, P.O. Box 5969, Safat, Kuwait
[b]Department of Management, Ball State University, Muncie, IN 47306, USA

Received 1 September 1997; received in revised form 28 May 1998; accepted 1 June 1998

Abstract

The majority of scheduling research assumes setup as negligible or part of the processing time. While this assumption simplifies the analysis and/or reflects certain applications, it adversely affects the solution quality for many applications which require explicit treatment of setup. Such applications, coupled with the emergence of production concepts like time-based competition and group technology, have motivated increasing interest to include setup considerations in scheduling problems. This paper provides a comprehensive review of the literature on scheduling problems involving setup times (costs). It classifies scheduling problems into batch and non-batch, sequence-independent and sequence-dependent setup, and categorizes the literature according to the shop environments of single machine, parallel machines, flowshops, and job shops. The suggested classification scheme organizes the scheduling literature involving setup considerations, summarizes the current research results for different problem types, and finally provides guidelines for future research.

Keywords: Machine and shop scheduling; Setup times; Setup cost; Review of problems and algorithms

1. Introduction

Setup includes work to prepare the machine, process, or bench for product parts or the cycle [1]. This includes obtaining tools, positioning work in process material, return tooling, cleanup, setting the required jigs and fixtures, adjusting tools, and inspecting material. The setup operation (time and/or cost) has for long been considered negligible and hence ignored, or considered as part of the processing time for the case of setup time. While this may be justified for some scheduling problems, many other situations call for explicit (separable) setup time (cost) consideration. For a separable setup, two problem types exist. In the first type, setup depends only on the job to be processed, hence it is called *sequence-independent*. In the second, setup depends on both the job to be processed and the immediately preceding job, hence it is called *sequence-dependent*.

The importance of setup times has been investigated in several studies. Wilbrecht and Prescott [2] found that sequence-dependent setup times are significant when a job shop is operated at or near full capacity. In a survey of industrial managers, Panwalkar et al. [3] discovered that about three quarters of the managers reported at least some operations they schedule require sequence-dependent setup times, while approximately 15% reported all operations requiring sequence-dependent setup times. Flynn [4] determined that applications of both sequence-dependent setup procedures and group technology principles increase output capacity in a cellular manufacturing shop, and Wortman [5] underlined the importance of considering sequence-dependent setup times for the effective management of manufacturing capacity. Krajewski et al. [6]

* Corresponding author. Tel.: +965-481-1188/7874; Fax: +965-484-7893; E-mail: allahverdi@kuc01.kuniv.edu.kw.
[1] Fax: +1-765-2858024; E-mail: jgupta@gw.bsu.edu
[2] Fax: +965-484-7893; E-mail: tariq@kuc01.kuniv.edu.kw

0305-0483/99/$ - see front matter
PII: S0305-0483(98)00042-5

examined the factors in a production environment that have the biggest influence on performance and concluded that, regardless of the production system in use, simultaneous reduction of setup times and lot sizes is the most effective way to reduce inventory levels and improve customer service. While recent advances in manufacturing technologies such as Flexible Manufacturing Systems (FMS) or single-minute exchange of die (SMED) concepts have reduced the influence of setup time, there are still many environments where setup time is significant.

Furthermore, treating setup time separately from processing time allows operations to be performed simultaneously and hence improves performance. This concept is inherent in recent production management philosophies and techniques such as just-in-time (JIT), optimized production technology (OPT), group technology (GT), cellular manufacturing (CM), and time-based competition [7–11].

In some situations setup cost is directly proportional to setup time. This is typically true when concern is limited only to machine idle time. However, there are other situations, where the cost is relatively high for switching between certain jobs even though their switching time is relatively less. This may be attributed in some situations to the high-skilled labor requirement. For the situation where setup cost and setup time are directly proportional, a schedule that is optimal with respect setup time is also optimal with respect to setup cost.

There are many practical applications that support the separate consideration of setup tasks from processing tasks. These applications can be found in various shop types and environments; e.g., production, service, and information processing. For sequence-independent setup applications, Bruno and Downey [12], Monma and Potts [13], and Chen [14] described a computer system application, in which computer jobs require different compilers. A setup is not incurred if the next job requires a compiler that is already resident in memory. However, if the next job requires a non-resident compiler, a setup time that depends only on the time needed to load the new compiler is incurred.

For sequence-dependent setup applications, Pinedo [15] described a paper bag factory where setup is needed when the machine switches between types of paper bags, and the setup duration depends on the degree of similarity between consecutive batches; e.g. size and number of colors. The printing industry provides numerous applications of sequence dependent setups where the machine cleaning depends on the color of the current and immediately following orders [16]. In several textile industry applications, setup for weaving and dying operations depends on the jobs sequence. In the container and bottle industry, the settings change depending on the sizes and shapes of the containers. Further, in the plastic industry, different types and colors of products require sequence-dependent setups [17, 18]. Similar practical situations arise in the chemical, pharmaceutical, food processing, metal processing, and paper industries [19–22]. Also, in an automatic turning center (ATC), the setup time depends on the difference in the number and type of the tools currently mounted on the turret and the ones required for the next workpiece. Other examples of sequence-dependent setup time applications include a semiconductor testing facility [23] and a machine shop environment [24]. Sule and Huang [24] described the activities typically associated with sequence-dependent and sequence-independent operations in machine shop environments.

Setup time (cost) problems can be classified into two levels; the first (lower) level, which is part of a shop floor control system, is concerned with the role of setup in scheduling jobs with known lot (batch) sizes on one or more machines to optimize certain objectives [22]. The second level is concerned with the role of setup in decision making at the strategic business planning level in production planning and control (e.g. aggregate production planning, MRP, inventory planning), where typical outputs of such decisions include the determination of lot sizes and order release dates [25, 26]. In order to limit its size and to provide adequate focus, this paper does not consider problems in the second level or problems in the first level which are related to lot streaming (transfer batches). For recent references related to lot streaming work with setup considerations, see [27–34].

Several survey articles have been written on various subjects in scheduling, and some researchers have written more than one survey paper on the same subject; e.g. [35, 36]. However, only a decade ago, Gupta and Kyparisis' survey of single-machine scheduling problems [37] cited only 13 references involving setup time (cost) considerations. Even though scheduling research involving setup considerations began in the mid-sixties and has grown tremendously in the last decade, no comprehensive review paper has appeared on the subject. Therefore, the objective of this paper is to review the literature related to static shop scheduling problems involving setup considerations where a shop may be characterized as a single-machine, parallel-machine shop, flowshop, or a job shop.

The rest of the paper is organized as follows: Section 2 presents the notations and a classification schema for the static scheduling problems involving setup considerations. Sections 3–6 review the setup literature in the single-machine, flow-shop, parallel machine, and job shop environments respectively. Finally, Section 7 summarizes the current status of scheduling research involving setup considerations and provides guidelines for future research.

2. Notation and classification

Setup is sequence-dependent if its duration (cost) depends on both the current and the immediately preceding job, and is sequence-independent if its duration (cost) depends only on the current job to be processed. A further classification categorizes setup problems as *batch* or *non-batch*. A *batch setup problem* occurs when part types are grouped into batches (or product families) and a (major) setup time is incurred when switching between part types belonging to different batches, and, in some applications, a (minor) setup is incurred for switching between part types within batches (i.e. from the same product families). In other words, a major setup time depends only on the batch being switched to, and the minor setup time depends only on the part type being switched to [38]. Fig. 1 shows the classification of setup problems adopted in this survey paper. This classification is applied to each shop environment: single machine, parallel machines, flowshop, and job-shop.

Following Lawler et al. [39], the standard three-field notation is used to describe a scheduling problem. The first field describes the shop type, the second field is reserved for setup information and other shop conditions, while the third field contains the performance criteria.

Shop type

1: single machine
F_m: flowshop with m machines
P: identical parallel machines
Q: uniform parallel machines
R: unrelated parallel machines
J: job shop
O: open shop

Shop conditions

r_j: non-zero job ready time or dynamic release date
nwt: no-wait
zbfr: zero-buffer
block: blocking
prmp: preemptive
prec: precedence constraint
brkdwn: machine breakdowns

Setup information

ST_{si}: sequence-independent setup time
SC_{si}: sequence-independent setup cost
ST_{sd}: sequence-dependent setup time
SC_{sd}: sequence-dependent setup cost
$ST_{si,b}$: sequence-independent batch setup time
$SC_{si,b}$: sequence-independent batch setup cost
$ST_{sd,b}$: sequence-dependent batch setup time
$SC_{sd,b}$: sequence-dependent batch setup cost
R_{si}: sequence-independent removal time
R_{sd}: sequence-dependent removal time

Performance criteria

C_{max}: makespan
L_{max}: maximum lateness
T_{max}: maximum tardiness
TSC: total setup/changeover cost
TST: total setup/changeover time
TFT: total flowtime
TCT: total completion time
TE: total earliness
TT: total tardiness
MFT: mean flowtime
MCT: mean completion time
MT: mean tardiness
NLJ: number of late (tardy) jobs
WTE: weighted total earliness
WTT: weighted total tardiness
WMFT: weighted mean flow time
WTFT: weighted total flowtime
WMCT: weighted mean completion time
WTCT: weighted total completion time

3. Single-machine problems

Arriving tasks (jobs) require service from a single available resource. Tasks are performed by the resource one at

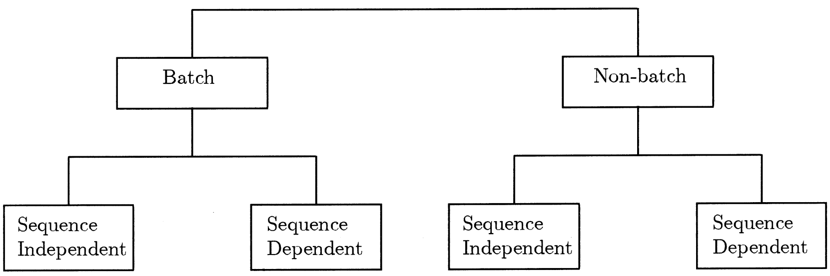

Fig. 1. Scheduling problems with setups.

a time and each task has different properties; e.g. processing time, due date, setup time, and weight. Job sequences may be determined to satisfy a certain criteria based on different performance measures. This section addresses the setup literature in a single-machine environment. Tables 1 and 2 summarize the non-batch and batch setup literature respectively.

3.1. Sequence-independent non-batch setups

For the regular single-machine problem, there is no point in explicitly considering setup times separate from processing times if setup times are sequence-independent. However, for certain cases (e.g. preemptive scheduling), considering them as separate could easily be justified.

For the case where changeover cost is constant, Glassey [40] proposed a dynamic programming (DP) algorithm to minimize the number of changeovers given that each job has to meet its delivery schedule. Mitsumori [41] extended Glassey's problem for the non-constant changeover cost case. Miyazaki and Ohta [42] provided an optimal algorithm for a modified flowtime criterion which is defined to be the elapsed time of each job counted from the start time to the common due date. In the preemption case,

Table 1
Single-machine scheduling research with non-batch setups

Setup type	Author(s) and Refs.	Criterion (additional conditions)
ST_{si}/SC_{si}	Glassey [40]	number of changeovers (deadlines)
	Mitsumori [41]	TSC (deadlines)
	Miyazaki and Ohta [42]	modified TFT (common due date)
	Zdrzalka [43]	maximum delivery time (r_j; prmp)
ST_{sd}/SC_{sd}	Gilmore and Gomory [46]	TSC
	Presby and Wolfson [47]	TSC
	Driscoll and Emmons [48]	TSC (with due dates)
	Burstall [49]	cost function
	Buzacott and Dutta [50]	cost function
	Taha [51]	cost function
	Sielken [52]	cost function
	Barnes and Vanston [53]	TSC and linear delay penalties
	Kusiak and Finke [54]	TSC (prec)
	Laguna et al. [55]	TSC
	Laguna and Glover [56]	TSC
	Feo et al. [57]	TSC
	Kolahan et al. [58]	cost function
	Gavett [59]	TST
	Haynes et al. [60]	TST
	White and Wilson [61]	TST
	Woodruff and Spearman [62]	TST (due date feasible)
	Farn and Muhlemann [63]	TST (r_j)
	Lockett and Muhlemann [64]	TST
	Moore [65]	selecting optimal subset of jobs during a fixed period
	Arcelus and Chandra [66]	MFT
	Miyazaki and Ohta [42]	modified TFT
	Bianco et al. [20]	C_{max} (r_j)
	He and Kusiak [67]	C_{max} (prec)
	Picard and Queyranne [68]	TT and WTT
	Uzsoy et al. [69]	L_{max} (prec)
	Uzsoy et al. [70]	L_{max}, NLJ (prec, r_j)
	Ovacik and Uzsoy [71]	L_{max} (r_j)
	Ovacik and Uzsoy [72]	L_{max} (r_j)
	Rubin and Ragatz [73]	TT
	Kim et al. [74]	WTT
	Lee et al. [75]	WTT
	Raman et al. [76]	WTT
	Asano and Ohta [77]	TE (r_j)
	ten Kate et al. [78]	f(WTT, WTE, TSC)
	Tan and Narasimhan [79]	WTT + TSC

Table 2
Single-machine scheduling research with batch setups

Setup type	Authors and Refs.	Criterion (additional conditions)
$ST_{si,b}/SC_{si,b}$	Bruno and Downey [12]	TST, TSC (batch deadlines)
	Bruno and Sethi [80]	Total holding cost, also prec
	Herrmann and Lee [82]	TSC (fam. dead lines)
	Mason [83]	TSC
	Ozden et al. [85]	TSC
	Liao and Liao [86]	TCT
	Gupta et al. [87]	(TCT, Range; hierarchical)
	Liao [88]	(TCT, Range)
	Zdrzalka [89, 90]	time by which all jobs delivered
	Baker [11]	$L_{\max}$ (constant setup)
	Baker and Magazine [91]	$L_{\max}$ (constant setup)
	Ghosh and Gupta [92]	$L_{\max}$
	Constantopoulos [93]	TCT, $L_{\max}$
	Williams and Wirth [94]	TCT
	Crauwels et al. [95]	WTCT
	Schutten et al. [96]	$L_{\max}$ (r_j)
	Hariri and Potts [97]	$L_{\max}$
	Unal and Kiran [98]	certain production requirement
$ST_{sd,b}/SC_{sd,b}$	Sahney [99]	MFT
	Gupta [100]	MFT (two batches)
	Potts [101]	TCT, WTCT (two batches)
	Woodruff and Spearman [62]	cost function (two batches)
	Gupta [102]	MFT
	Mason and Anderson [81]	WMFT (additive setup)
	Psaraftis [103]	cost function (job priority)
	Ozgur and Brown [104]	$C_{\max}$
	Raman et al. [76]	MFT, MT (r_j)
	Sawicki [105]	MT (random arrival)
	Monma and Potts [13]	$L_{\max}$, NLJ, WTCT
	Ovacik and Uzsoy [23]	$L_{\max}$
	Ahn and Hyun [106]	TCT
	Ghosh [107]	WTCT
	Chen [108]	total earliness and tardiness
	Unal et al. [10]	$C_{\max}$, WTCT (rescheduling)

Zdrzalka [43] proposed a heuristic solution with a specified worst-case performance ratio for minimizing the maximum delivery time in the presence of release dates.

3.2. Sequence-dependent non-batch setups

Unlike the sequence-independent setup time problem in which makespan is the same regardless of the selected sequence, when setup times are dependent on the sequence minimizing makespan becomes equivalent to minimizing the total setup time which corresponds to what is usually called the *traveling salesman problem* (TSP). In a TSP, each city corresponds to a job and the distance between cities corresponds to the time required to change from one job to another. If the setup times for all pairs of jobs are indifferent to their sequencing order when scheduled consecutively, the scheduling problem is equivalent to a *symmetrical* TSP, otherwise, it is equivalent to an *asymmetrical* TSP. Recent developments in solving TSP are summarized by Laporte [44] and Junger et al. [45].

One of the pioneering works on the sequence-dependent setup cost problem is by Gilmore and Gomory [46] who modeled and solved the problem for the total setup cost objective as a TSP. To solve the same problem, Presby and Wolfson [47] provided an optimization algorithm that is suitable only for small problems. Driscoll and Emmons [48] also addressed the same problem with deadline restriction on all jobs using Dynamic Programming (DP).

For a cost function which includes total setup cost, Burstall [49] developed a heuristic solution where branch-and-bound is used as a search procedure. Buzacott and Dutta [50] proposed a DP algorithm to minimize the overall operating cost which includes sequence-dependent changeover cost for a multi-purpose facility. An approximate solution was also developed for large problems. Taha [51] formulated the problem to include sequence-dependent setup times and costs, inventory cost, and penalty cost for finishing a job late. The formulation, which is based on a 0–1 polynomial programming, has the drawback of being computationally intractable. Sielken [52] reformulated Taha's problem as a smaller linear 0–1 mixed integer programming problem.

Barnes and Vanston [53] used a DP and a branch-and-bound algorithm to solve the $1|SC_{sd}|TSC$ problem with an additional criterion of linear delay penalties. For the same problem with precedence constraint, Kusiak and Finke [54] used a network formulation to develop a branch-and-bound algorithm. For the $1|SC_{sd}|TSC$ problem, Laguna et al. [55] presented a tabu search method while Laguna and Glover [56] used an integrated target analysis and tabu search method. Feo et al. [57] developed a greedy randomized data adaptive search procedure and showed it to be competitive with other methods with respect to the solution values obtained and the CPU time required. Kolahan et al. [58] used a tabu search approach to find a good sequence such that, once the tool replacement decision has been made in an automated machine center, the total expected production cost which includes sequence-dependent setup cost is minimized.

Gavett [59], Haynes et al. [60], and White and Wilson [61] provided several heuristic rules for the $1|ST_{sd}|TST$ problem. White and Wilson [61] also developed a procedure that classifies setup operations and forecasts the setup times. Woodruff and Spearman [62] developed a branch-and-bound algorithm that finds the due date feasible sequence with the minimum setup time. Farn and Muhlemann [63] also considered the sequence-dependent setup time problem in the presence of dynamic job arrivals and established that the best heuristic for the static problem is not necessarily the best in a dynamic situation. Lockett and Muhlemann [64] used a branch-and-bound algorithm and heuristics to solve the sequence-dependent setup time problem on a single machine with multiple tools. Since, each tool change takes the same amount of time, their objective of minimizing the total number of tool changes is equivalent to the objective of minimizing total changeover time. Moore [65] developed an algorithm that selects an optimal subset of available jobs to be performed during a fixed period of time.

Arcelus and Chandra [66] and Miyazaki and Ohta [42] addressed the $1|ST_{sd}|MFT$ problem. Miyazaki and Ohta [42] defined mean flowtime as the elapsed time of each job counted from the start time to the common due date.

Bianco et al. [20] formulated the $1|r_j,ST_{sd}|C_{\max}$ problem as a mixed integer linear programming (MILP), and developed a heuristic algorithm using lower bounds and dominance criteria. For the $1|$prec, $ST_{sd}|C_{\max}$ problem, He and Kusiak [67] proposed a simpler mixed integer formulation and a fast heuristic algorithm of low computational time complexity.

Picard and Queyranne [68] modeled the single-machine problem as a time-dependent traveling salesman problem and developed an optimization method that uses a branch-and-bound algorithm to minimize total tardiness and weighted total tardiness. Uzsoy et al. [69] provided a branch-and-bound algorithm to find an optimal solution for the $1|$prec, $ST_{sd}|L_{\max}$ problem. The algorithm, however, is cumbersome for a large number of jobs (more than 15 jobs). For dynamic job arrivals, Uzsoy et al. [70] proposed heuristics for the same problem, and in addition addressed the $1|ST_{sd}|NLJ$ problem. Ovacik and Uzsoy [71] solved the $1|r_j$, $ST_{sd}|L_{\max}$ problem by using information on jobs that will become available over a given forecast window, while Ovacik and Uzsoy [72] presented a family of rolling horizon heuristics for the $1|r_j,ST_{sd}|L_{\max}$ problem. These heuristics were shown to outperform myopic dispatching rules by an order of magnitude in very reasonable computation times.

Rubin and Ragatz [73] used a genetic search algorithm to solve the $1|ST_{sd}|TT$ problem. When compared with a branch-and-bound algorithm, their proposed algorithm was found to be competitive in some problems. Kim et al. [74] used a hybrid approach that combines dispatching rules and neural networks to solve the $1|ST_{sd}|WTT$ problem. Lee et al. [75] proposed a three-phase heuristic solution procedure that uses the Apparent Tardiness Cost with Setups (ATCS) rule to solve the $1|ST_{sd}|WTT$ problem and showed that it outperforms the rule suggested by Raman et al. [76]. Asano and Ohta [77] developed a branch-and-bound algorithm combined with dominance relations to find an optimal solution for the $1|r_j$, $ST_{sd}|TE$ problem. A mixed integer programming formulation and heuristic algorithms for solving the weighted earliness, tardiness, and setup costs are described by ten Kate et al. [78]. Tan and Narasimhan [79] used simulated annealing to minimize the sum of weighted tardiness and total setup cost.

3.3. Sequence-independent batch setups

Bruno and Downey [12] considered the $1|ST_{si,b}|TST$ and $1|SC_{si,b}|TSC$ problems where job batches have deadlines. They demonstrated that even very restrictive cases of the problem are NP-hard. Without due dates and for both cases of precedence and non-precedence constraints, Bruno and Sethi [80] used a dynamic programming approach to solve the same problem but with the objective of minimizing total holding cost.

In Bruno and Sethi's [80] model jobs are ready at time zero and there are no due dates. Their problem can be solved by minimizing total weighted flowtime where the holding costs are used as job weights [81]. A genetic algorithm for the $1|SC_{si,b}|TSC$ problem with family deadlines is developed by Herrmann and Lee [82] while Mason [83] described a genetic algorithm for the $1|SC_{si,b}|TSC$ problem. Major developments in single machine scheduling problems with group technology assumption and sequence independent setup times are summarized by Webster and Baker [84].

Ozden et al. [85] developed a dynamic programming based algorithm to solve the $1|SC_{si,b}|TSC$ problem when both major and minor setups are present. A heuristic solution procedure for the $1|SC_{si,b}|TCT$ problem is developed by Liao and Liao [86]. Gupta et al. [87] showed that the problem of hierarchically minimizing range of completion times and makespan in the presence of customer orders for jobs from various batches requiring setup times can be polynomially solved. Liao [88] developed a branch and bound algorithm for finding all efficient solutions for the above problem considered by Gupta et al. [87].

Zdrzalka [89] provided three approximate algorithms with worst-case performance bounds to solve the problem with unit batch setup times, where the objective is to minimize the time by which all jobs are delivered. He pointed out that this problem is equivalent to minimizing maximum lateness with a modified due date. Zdrzalka [90] extended his previous work to non-unit batch setup times and proposed two approximate algorithms with worst-case performance ratio and solution time.

Baker [11] and Baker and Magazine [91] considered the $1|ST_{si,b}|L_{\max}$ problem for a constant batch setup. While Baker [11] developed two heuristic procedures for the problem that use neighborhood search routines to improve an existing heuristic, Baker and Magazine [91] provided an algorithm that uses a branch-and-bound procedure combined with dominance properties to solve the problem. Ghosh and Gupta [92] developed an improved dynamic program for finding an optimal solution for the $1|ST_{si,b}|L_{\max}$ problem.

Constantopoulos et al. [93] addressed the problems of $1|ST_{si,b}|TCT$ and $1|ST_{si,b}|T_{\max}$ using a dynamic programming formulation for the former, and a branch-and-bound for the latter. A heuristic for the $1|ST_{si,b}|TCT$ problem is developed by Williams and Wirth [94]. Crauwels et al. [95] considered the same problem with the objective of weighted total completion time. They investigated four local search heuristics: multi-start descent, simulated annealing, threshold accepting, and tabu search. All four heuristics were reported to yield less than 0.4% deviation from the optimal solution for up to 50 jobs. A superior performance was also obtained with a hybrid method that uses tabu search and genetic algorithm.

For the maximum lateness objective and non-zero ready times, Schutten et al. [96] used a branch-and-bound algorithm to optimally solve problems with up to fifty jobs. Hariri and Potts [97] also developed a branch-and-bound algorithm where all jobs are ready at time zero for the same problem. Their algorithm optimally solves problems with up to sixty jobs. They also developed a heuristic for the case where a batch may be divided into at most two groups.

Unal and Kiran [98] considered the problem of sequencing batches with batch setups where the objective is to determine the availability of a sequence that satisfies production requirements for each part type at a specified time.

3.4. Sequence-dependent batch setups

Sahney [99] presented a branch-and-bound algorithm to deal with the problem of switching a server between two parallel machines in order to minimize mean flow time. This problem is equivalent to a single-machine problem with two batches [100]. For two batches, Gupta [100] developed a polynomially bounded solution algorithm with respect to the mean flowtime criterion. However, Potts [101] provided a counter-example to Gupta's algorithm and proposed a DP for both objectives of total completion time and weighted total completion time. Also, for two batches, Woodruff and Spearman [62] developed a solution algorithm based on the method of tabu search with considerations of setup times, setup costs, holding costs, and deadlines. Gupta [102] provided a heuristic solution to solve the problem of Gupta [100] for m batches. Notice that when there are only two batches, sequence-dependent and sequence-independent are equivalent problems provided that setup is only incurred when switching between batches.

Mason and Anderson [81] addressed the $1|ST_{sd,b}|WMFT$ problem in which setup times are additive (see Section 5.2 for additive setup definition). Except for one theorem, they considered the second term in the additive model to be zero. This consideration makes their setup model equivalent to the sequence-independent setup time model as in Williams and Wirth [94]. Although Mason and Anderson's [81] paper is more general than Gupta's paper [102] in its objective criterion, Gupta's is more general in its setup time structure.

Psaraftis [103] developed a DP algorithm with priority consideration to minimize a total processing cost function, of which changeover cost is included. The proposed algorithm can also be used to minimize total completion time and total weighted completion time. Ozgur and Brown [104] described a two-phase TSP based procedure for the problem of sequence dependent batch scheduling problem to minimize makespan.

Raman et al. [76] investigated the dynamic scheduling of an automated manufacturing center in which jobs are processed in batches and constant setup times between batches are incurred. For the mean flow time and mean tardiness criteria, they developed the characteristics of the optimal solution. They also provided an implicit enumeration approach to solve the problem with the mean tardiness objective. Also for the mean tardiness objective, Sawicki [105] used a simulation model to compare different rules where job arrivals occur at random (inter-arrival time is geometrically distributed).

Monma and Potts [13] considered batch sequence-dependent setup time under the assumption of triangular inequality. Using dynamic programming algorithms, they showed the problems with the objectives of maximum lateness, total weighted completion time, and number of late jobs to be efficiently solvable for a fixed number of batches. However, they noted that their algorithms are largely theoretical. Ovacik and Uzsoy [23] addressed the $1|ST_{sd,b}|L_{max}$ problem and showed that the addition of a local improvement procedure results in substantial improvements over the schedules obtained by dispatching rules alone.

Ahn and Hyun [106] solved the $1|ST_{sd,b}|TCT$ problem using a DP approach. For the weighted total completion time criterion, Ghosh [107] provided a dynamic programming formulation and showed that, under the triangular inequality assumption for setup time, the problem is strongly NP-hard even when the jobs within a batch have the same processing time and weight.

Chen [108] provided a polynomial DP algorithm in the case of earliness–tardiness penalties for two criteria. The first criterion minimizes the total earliness and tardiness penalties, while the second extends the first criterion to include the total due date penalty.

A recent trend in the scheduling problems, called rescheduling, involves updating a schedule in the face of new arriving jobs without disturbing the commitment of the jobs in the already existing schedule. This problem is particularly important in the make-to-order production environment. For the new arriving jobs, Unal et al. [10] provided a polynomial-time algorithm for the C_{max} criterion. They also showed the rescheduling problem for the WTCT criterion to be strongly NP-hard and developed two heuristics with data-dependent worst-case error bounds.

So far we have considered the problem with a single machine. In some cases there is more than one machine available for performing the same job; this is called parallel-machine problem and it is covered in the next section.

4. Parallel-machine problems

Arriving jobs can be processed on any one of a number of available machines. Each job, with different characteristics, has a single operation that can be performed on any machine. Job schedules may be determined to satisfy a certain criteria based on different performance measures. This section describes the setup literature in a parallel-machine environment; a summary of which is provided in Table 3.

4.1. Sequence-independent batch setups

No results are known for parallel-machine scheduling problems with sequence independent non-batch setups. For the parallel-machine problem with sequence independent batch setup times, Bruno and Sethi [80] used DP to minimize the total holding cost in the case of batch setup time for the identical parallel-machine problem. For the makespan criterion, Tang [38] and Wittrock [109], and Wittrock [110] solved the problem using two heuristic algorithms.

Monma and Potts [13] considered the problem of two identical parallel machines with batch setup times. For both preemptive and non-preemptive cases with fixed number of batches, they established that pseudo-polynomial algorithms exist for the performance measures of makespan, maximum lateness, number of late jobs, and total weighted completion time. They also showed these problems to be NP-hard when the number of batches is arbitrary. The problem with the total completion time criterion was left as an open problem. However, Cheng and Chen [111] later showed that this problem is binary NP-hard even for equal processing times. For the makespan objective and preemptive case, Monma and Potts [112] extended their earlier work to the m-machine problems and provided two types of heuristics and worst-case performance ratios. These heuristics were modified by Chen [14] to yield a better worst-case performance ratio with the same time requirement.

So [113], Tang [38], and Wittrock [114] considered the parallel identical machines scheduling problem with minor and major setups for types and families of jobs respectively. So [113] developed three heuristics to maximize a total reward function under the assumption of fixed processing machine capacity. Although So [113] considered total reward as the performance measure, he stated that the developed heuristics can be modified to minimize makespan. Tang [38], on the other hand, provided lower bounds and heuristics to solve the static problem with the objective of minimizing makespan. Wittrock [114] improved Tang's [38] results by developing an iterative greedy heuristic which utilized the major and minor setups in sorting job-families and part types within a family respectively.

Table 3
Parallel machine scheduling research with setups

Setup type	Author(s) and Refs.	Criterion (additional conditions)
$ST_{si,b}/SC_{si,b}$	Bruno and Sethi [80]	cost function
	Tang and Wittrock [109]	C_{max}
	Wittrock [110]	C_{max}
	Monma and Potts [13]	C_{max}, L_{max}, NLJ, WTCT (two-machine, prmp and non-prmp)
	Cheng and Chen [111]	TCT (two machines)
	Monma and Potts [112]	C_{max} (prmp)
	Chen [14]	C_{max} (prmp)
	So [113]	total reward (minor and major setups, fixed processing capacity)
	Tang [38]	C_{max} (minor and major setups)
	Wittrock [114]	C_{max} (minor and major setups)
	Dietrich [115]	$f[C_{max}$, TCT] (minor and major setups)
	Schutten and Leussink [116]	L_{max} (r_j)
	Brucker et al. [117]	L_{max} (r_j, deadlines)
ST_{sd}/SC_{sd}	Marsh and Montgomery [118]	TSC (identical and non-identical)
	Deane and White [119]	TSC (workload balancing)
	Geoffrion and Graves [120]	cost function
	Guinet [121]	TFT, T_{max}
	Elmaghraby et al. [122]	TFT, T_{max}
	Parker et al. [123]	TSC (workload capacity)
	Dearing and Henderson [124]	TSC
	Sumichrast and Baker [125]	TST
	Ovacik and Uzsoy [126]	L_{max}, C_{max} (bounded setup time)
	Franca et al. [18]	C_{max}
	Ovacik and Uzsoy [127]	L_{max} (r_j)
	Guinet and Dussauchoy [128]	C_{max}, L_{max}
	Lee and Pinedo [129]	WTT
$ST_{sd,b}/SC_{sd,b}$	Bitran and Gilbert [21]	TSC (minor and major setups)
	Ghosh [107]	TCT, WTCT

Dietrich [115] considered the unrelated parallel machine problem with major and minor setups and developed a two-phase heuristic for minimizing a linear function of the schedule makespan and total flow time.

For the maximum lateness criterion, Schutten and Leussink [116] developed a branch-and-bound algorithm to solve the identical parallel machine problem where jobs have release dates, due dates, and batch setups. They presented two lower bounds, one of which allows preemption. For establishing the lower bounds, setup times are modeled as setup jobs with release dates, due dates, and processing times. Brucker et al. [117] considered the above problem with deadlines and developed fully polynomial approximation scheme for its solution.

4.2. Sequence-dependent non-batch setups

Marsh and Montgomery [118] proposed heuristic approaches to address the sequence-dependent changeover times on parallel machines for the two cases of identical and non-identical machines with respect to minimizing setup time. For non-identical machines, changeover times depend on both the sequence and the machine. Deane and White [119] extended Marsh and Montgomery results to include workload balancing. They provided a branch-and-bound algorithm to minimize sequence-dependent setup cost. Geoffrion and Graves [120] developed a quadratic assignment formulation of the problem to minimize the sum of the changeover, production, and time-constraint penalty costs. Guinet [121] used the mixed integer programming formulation to develop heuristic algorithms for the $R|ST_{sd}|$MFT and $R|ST_{sd}|$MFT problems. Elmaghraby et al. [122] extended Guinet's [121] results to develop improved iterative heuristic to minimize makespan.

Parker et al. [123] formulated capacity-restricted parallel-machine problem with sequence-dependent changeover costs as a vehicle routing problem and described algorithms to minimize the total changeover costs. For the criterion of total setup cost, Dearing and Henderson [124] used integer linear programming to model the problem in a textile weaving application. Sumichrast and Baker [125] proposed a heuristic method based on integer programming that improves Dearing and Henderson's [124] results.

Ovacik and Uzsoy [126] studied the $P|ST_{sd}|C_{max}$ and $P|ST_{sd}|L_{max}$ problems in semiconductor testing facilities where setup times are bounded by processing times. They provided an example showing that list schedules are non-dominant, and developed worst-case error bounds for list scheduling algorithms. Franca et al. [18] considered the same problem of Ovacik and Uzsoy [126] for the makespan objective with no restriction on setup time and developed a three-phase heuristic which uses a tabu search method. For the dynamic version of Franca et al. [18] problem, Ovacik and Uzsoy [127] provided a family of rolling horizon heuristics to minimize maximum lateness. The proposed heuristics outperform dispatching rules even when combined with local search methods. Guinet and Dussauchoy [128] used an extension of the Hungarian method to solve the linear assignment problem as a heuristic to solve the $P|\mathrm{ST}_{\mathrm{sd}}|C_{\max}$ problem.

Lee and Pinedo [129] provided a three-phase heuristic that combines a dispatching rule and a simulated annealing procedure to solve the $P|\mathrm{ST}_{\mathrm{sd}}|\mathrm{WTT}$ problem. The rule is a modification of the ATCS rule used for the single machine by Lee et al. [75].

4.3. Sequence-dependent batch setups

Bitran and Gilbert [21] developed heuristic methods to solve the $P|\mathrm{SC}_{\mathrm{sd,b}}|\mathrm{TSC}$ problem which has two magnitudes of setup costs; one being much larger than the other. They pointed out that the minimization of the considered criterion is roughly equivalent to minimizing the number of large setups. Ghosh [107] addressed the $P|\mathrm{ST}_{\mathrm{sd,b}}|\mathrm{TCT}$ and $P|\mathrm{ST}_{\mathrm{sd,b}}|\mathrm{WTCT}$ problems where batch setup times satisfy the triangular inequality. For an arbitrary number of batches or machines, the problem was shown to be strongly NP-hard with respect to weighted completion time. When only the number of batches is arbitrary and jobs are identical, the problem is still strongly NP-hard. However, for fixed number of batches, the total completion time problem is polynomially solvable.

In a single and parallel machine environments, a job has only one operation. It might be that a job requires more than one operation and the operations need to be performed by different machines. The following two sections address this problem for different shop environments.

5. Flowshop problems

In an m-machine flowshop, arriving jobs have to be processed on m machines in the same technological order. Each job has m operations and operation i of each job is to be performed on machine i. Operation times for each job on different machines may be different.

The flowshop problem was first studied by Johnson [130] for two machines with the objective of minimizing makespan. All subsequent flowshop models are extension of Johnson's work. For the classical two-machine flowshop, permutation schedules are dominant with respect to any regular measure; i.e. in order to find an optimal solution one only needs to consider the same sequence of jobs on both machines.

When setup times are separable and sequence-dependent on only one machine or sequence-independent on both machines, this property still holds [131, 132]. However, when setup times are sequence-dependent on both machines, this property does not hold [133]. Furthermore, permutation schedules are not dominant with respect to any regular measure if both time lags and setup times are considered. Start (stop) lag of a job is defined to be the minimum time that must elapse between starting (completing) the job on the current machine and starting (completing) it on the following machine [134, 135].

Depending on the completion time definition, permutation schedules may or may not be dominant for the case of sequence-independent setup times when removal times are also separate and sequence-independent [136]. If the job completion time does not

Table 4
Permutation schedule dominance for the $F2|S_{\mathrm{si}},R_{\mathrm{si}}|\gamma$ problem

Prob.	Setup	Removal		Time lags	Permutation schedule	
		mach-1	mach-2		dominant	non-dominant
1	✓				✓	
2	✓			✓		✓
3		*j*-based	*j*-based			✓
4		*j*-based	*m*-based			?
5		*m*-based	*m*-based		✓	
6	✓	*j*-based	*m*-based			✓
7	✓	*m*-based	*m*-based		✓	
8	✓	*m*-based	*j*-based		✓	

include its removal time, it is called job-based (j-based) completion time [137]. When it includes removal time, as defined by Sule [138] and Sule and Huang [24], it is called machine-based (m-based) completion time. Table 4 summarizes the permutation dominance property for certain types of two-machine flowshops.

In the classical flowshop, an infinite buffer is assumed and jobs may wait on or between the machines. These assumptions may not hold in some flowshops. In a no-wait (nondelay) flowshop scheduling problem, operations of a job have to be processed from start to end without interruptions either on or between machines [139]. For the classical flowshop, where an infinite-buffer exists between the machines with no restriction on the in-process waiting times, finding an optimal schedule with respect to any regular measure requires no inserted idle time on the first machine. However, for the case of no-wait, we might have some idle time on all machines.

A machine may be blocked when the buffer capacity is full. A zero-buffer problem occurs when there is no intermediate buffers between machines. This is different from the no-wait problem where a job processing cannot be interrupted on or between machines [139]. The zero-buffer and no-wait problems are equivalent for the two-machine problem when setup times are included in processing times. For separable setup times, however, there are two cases in the zero-buffer problem. In the first case, the setup of the next job on machine 1 is not allowed until the current job releases machine 1. In the second case, the setup for the next job on machine 1 can start as soon as machine 1 completes its processing of the current job. The first case seems to be more practical. Notice that the zero-buffer in the first case is equivalent to the no-wait problem [140].

Tables 5 and 6 summarize the results for flowshop problems involving setup considerations for the non-batch and batch categories respectively.

5.1. Sequence-independent non-batch setups

Yoshida and Hitomi [132] extended Johnson's [130] model to the case where setup times are separable from processing times and modified his algorithm to obtain an optimal solution for the two-machine flowshop with the objective of minimizing makespan. Khurana and Bagga [141] optimally solved the problem of Yoshida and Hitomi [132] subject to the constraint of completing the processing of a set of jobs before a given deadline. Allahverdi [142] relaxed the continuous machine availability assumption by taking into account machine breakdowns. He established a dominance relation for minimizing makespan with probability 1, and showed that Yoshida and Hitomi's [132] algorithm stochastically minimizes makespan under certain conditions.

Sule [138], and Sule and Huang [24] extended Yoshida and Hitomi's [132] model by considering not only setup but also removal times as separable from processing times, and provided an updated optimal algorithm for the problem.

Sule and Huang [24] also considered the three-machine flowshop problem for which two heuristic algorithms are provided. Allahverdi [143] considered the two-machine case problem in Sule and Huang [24] and showed that their algorithm stochastically minimizes makespan when machines suffer random breakdowns. Proust et al. [144] considered the $F_m|S_{si}, R_{si}|C_{max}$ problem and developed a branch-and-bound algorithm for small size problems and four heuristics to solve larger problems. A heuristic algorithm is also developed by Rajendran and Ziegler [145].

When time lags are considered in addition to setup times, Khurana and Bagga [146] provided an optimal solution for a two-machine flowshop to minimize makespan under two conditions: (i) the start and stop lags are equal for each job, and (ii) once the processing time of a job is greater (smaller) on machine 1 than its processing time on machine 2, then the sum of processing and setup times of this job on machine 1 is also greater (smaller) than that of the job on machine 2. Szwarc [147] used the formula he developed in a previous paper [148] to find an optimal solution for the same problem without the two conditions. In addition, he provided an approximate solution for the multiple machine case. Nabeshima and Maruyama [149] extended the two-machine problem of Szwarc [147] by also considering separate removal and transportation times. They additionally addressed the three-machine problem under certain conditions.

For the general m-stage permutation flowshop scheduling problem, Gupta [134] described dominance conditions and an optimization algorithm to minimize makespan. His algorithm is general enough to permit start and stop time lags in the problem as well. However, the proposed algorithm requires excessive computational effort and hence is not suitable to solve large-sized problems.

The papers discussed so far treat the flowshop problem with respect to the makespan criterion. Dileepan and Sen [150], however, considered the $F_2|ST_{si}|L_{max}$ problem, provided two sufficient conditions for an optimal solution, used these conditions to develop a branch-and-bound algorithm and tested the effectiveness of two heuristics to minimize maximum lateness. Allahverdi and Aldowaisan [151] extended Dileepan and Sen [150] work to include separate removal times. They found elimination criteria for the generic and ordered flowshops, and optimal solutions for special ordered flowshops.

Table 5
Flowshop scheduling research with non-batch setups

Setup type	Author(s) and Refs.	# Stages	Criterion (additional conditions)
ST_{si}/SC_{si}	Yoshida and Hitomi [132]	2	C_{max}
	Khurana and Bagga [141]	2	C_{max} (deadlines for sets of jobs)
	Allahverdi [142]	2	C_{max} (brkdwn)
	Sule [138]	2	C_{max} (R_{si})
	Sule and Huang [24]	2, 3	C_{max} (R_{si})
	Allahverdi [143]	2	C_{max} (R_{si}, brkdwn)
	Proust et al. [144]	m	C_{max} (R_{si})
	Rajendran and Ziegler [145]	m	C_{max} (R_{si})
	Khurana and Bagga [146]	2	C_{max} (time lag)
	Szwarc [147]	2, m	C_{max} (time lag)
	Szwarc [148]	2, m	C_{max} (R_{si}, time lag)
	Nabeshima and Maruyama [149]	2, 3	C_{max} (R_{si}, time lag)
	Gupta [134]	m	C_{max} (time lags)
	Dileepan and Sen [150]	2	L_{max}
	Allahverdi and Aldowaisan [151]	2	L_{max} (R_{si})
	Allahverdi [152]	2	MFT
	Gupta et al. [153]	2	C_{max} (nwt, R_{si})
	Aldowaisan and Allahverdi [140]	2	TFT (nwt)
	Cao and Bedworth [154]	m	C_{max} (non-zero transfer time)
	Park and Seudel [155]	m	C_{max} (limited buffer) time)
	Gupta and Tunc [137]	2, hybrid	C_{max} (multiple machines at stage 2)
ST_{sd}/SC_{sd}	Corwin and Esogbue [131]	2	C_{max} (setup dependent on one machine)
	Uskup and Smith [156]	2	TSC (with due dates)
	Gupta and Darrow [133]	2	C_{max}
	Gupta and Darrow [157]	2	C_{max}
	Gupta [158]	m	C_{max} (nwt, limited buffer)
	Gupta [159]	m	cost function
	Srikar and Ghosh [19]	m	C_{max}, MFT
	Stafford and Tseng [160]	m	C_{max}, MFT (nwt)
	Rios-Mercado and Bard [161]	m	C_{max}
	Gupta et al. [162]	m	C_{max}
	Rios-Mercado and Bard [163]	m	C_{max}
	Gupta [164]	m	C_{max}
	Das et al. [17]	m	C_{max}
	Simons [165]	m	C_{max}
	Rios-Mercado and Bard [166, 167]	m	C_{max}
	Parthasarathy and Rajendran [168]	m	WTT
	Parthasarathy and Rajendran [169]	m	WT_{max}
	Rajagopalan and Karimi [170]	m	C_{max} (non-zero transfer time, mixed storage policies)
	Szwarc and Gupta [171]	2, m	C_{max} (additive setup)

For the mean flowtime criterion, Allahverdi [152] addressed the two-machine flowshop problem with separable setup times. Since the problem is NP-hard even for non-separable setup times, he obtained optimal sequences for two special flowshops and a heuristic algorithm for the generic problem.

For the two-machine no-wait flowshop problem where setup and removal times are separable, Gupta et al. [153] found a polynomial time algorithm to minimize makespan. Aldowaisan and Allahverdi [140] considered the same problem with only setup time separated but with the objective of minimizing total flowtime. They developed optimal solutions for certain cases and a heuristic solution for the generic case. Their results apply to the zero-buffer case where the setup of the next job on the first machine is not allowed until the machine releases the current job.

The majority of research assumes transfer times to be zero. This assumption is valid as long as these times are very small with respect to processing times or they are large but equal for all jobs. Cao and Bedworth [154] provided a heuristic algorithm for the m-machine flowshop with non-zero transfer and setup times to minimize makespan. Since they consider the

Table 6
Flowshop scheduling research with batch setups

Setup type	Author(s) and Refs.	# Stages	Criterion (additional conditions)
$ST_{si,b}/SC_{si,b}$	Hitomi and Ham [173]	2	C_{max}
	Baker [172]	2	C_{max}
	Hitomi et al. [174]	m	C_{max}
	Ham et al. [175]	m	C_{max}
	Logendran and Sriskandarajah [9]	2	C_{max} (block, zbfr)
	Vakharia and Chang [176]	m	C_{max}
	Skiron-Kapov and Vakharia [177]	m	C_{max}
	Sridhar and Rajendran [178]	m	C_{max}
	Zdrzalka [179]	2	C_{max}, batch splitting
	Sotoskov [180]	m	C_{max}, batch splitting
	Dannebereg et al. [181]	m	C_{max}, batch splitting
	Li [182]	2, hybrid	C_{max} (multiple machine at stage 2)
$ST_{sd,b}/SC_{sd,b}$	Vakharia et al. [183]	m	C_{max}
	Schaller et al. [184]	m	C_{max}

removal time as part of the completion time of a job, the transfer operation time becomes identical to the removal time. Park and Steudel [155] proposed three heuristics and evaluated them against four existing heuristics for the permutation flowshop with separable sequence independent setup times and limited buffers between machines.

For a hybrid flowshop where the first stage consists of only one machine and the second consists of identical parallel machines, Gupta and Tunc [137] addressed the separable setup and removal times case to minimize makespan. When the number of machines at the second stage is greater than or equal to the number of jobs, they provided a polynomial optimization algorithm. For the opposite case, which is NP-hard, they proposed four heuristics.

5.2. Sequence-dependent non-batch setups

Corwin and Esogbue [131] addressed the two-machine flowshop problem with a makespan objective, where the setup times are sequence-dependent on the first (second) machine and sequence-independent on the second (first) machine. They showed the optimality of permutation schedules for both cases and established a dynamic programming formulation for each problem. Under the assumption that setup cost is directly proportional to setup time, Uskup and Smith [156] provided a branch-and-bound solution for the two machine flowshop problem with a total setup cost criterion such that the schedule meets all deadlines. Gupta and Darrow [133] generalized Corwin and Esogbue's [131] problem by considering sequence-dependent setup times on both machines for permutation schedules and established two heuristics for the problem. Gupta and Darrow [157] considered the same problem and showed it to be strongly NP-hard even when setup time is sequence-dependent on only one of the two machines for both permutation and non-permutation cases. They also specified conditions for which permutation schedules are dominant and provided four heuristics for both small and large size problems. Gupta [158] formulated the $F_m|ST_{sd}$, no-wait$|C_{max}$ problem as an asymmetrical TSP and showed that the flowshop scheduling problems with sequence dependent setup times are strongly NP-hard for the zero, limited, or infinite intermediate storage space cases.

Gupta [159] proposed a lexicographic search algorithm to solve the m-machine flowshop problem with an opportunity cost function criterion that includes sequence-dependent cost. The opportunity cost criterion is generic and other criteria such as makespan can be derived from it. Srikar and Ghosh [19] developed an MILP formulation for the m-machine problem to minimize makespan or mean flowtime. Stafford and Tseng [160] made some corrections to Srikar and Ghosh [19] paper, and developed three new MILP formulations for different problems including no-wait. For the makespan objective function, Rios-Mercado and Bard [161] developed several valid inequalities for two different MIP formulations of the $F_m|ST_{sd}|C_{max}$ problem. Gupta et al. [162] developed a branch and bound algorithm for the solution of the $F_m|ST_{sd}|C_{max}$ problem and solved problems involving 20 jobs. Rios-Mercado and Bard [163] empirically evaluated the MIP formulation based inequalities within a branch-and-cut framework and showed the effectiveness of this approach compared to the branch-and-bound algorithm.

Heuristic algorithms for the $F_m|ST_{sd}|C_{max}$ problem are developed by Gupta [164]. Das et al. [17] developed a time-saving index heuristic algorithm and showed that it gives good solutions when setup times are relatively larger than processing times. Heuristic algorithms for solving the $F_m|\mathrm{ST_{sd}}|C_{\max}$ problem are also developed by Simons [165] and Rios-Mercado and Bard [166, 167]. For the case where setup times are an order of magnitude smaller than processing times, Rios-Mercado and Bard's [167] algorithms are relatively superior to Simons' [165] SETUP heuristic. However, Simons' [165] SETUP heuristic is relatively superior for the case where both the setup and processing times are identically distributed. Parthasarathy and Rajendran [168] developed a simulated annealing algorithm to minimize mean weighted tardiness in a flowshop with sequence dependent setup times. Subsequently, they extended their heuristic to minimize the maximum weighted tardiness [169].

Rajagopalan and Karimi [170] presented a heuristic algorithm to minimize makespan in the m-machine flowshop with sequence-dependent setup times and non-zero transfer times and mixed-storage policies. Szwarc and Gupta [171] defined the additive model to be a special case of the sequence-dependent setup time model to minimize makespan for a two-machine flowshop. They divided setup time into two components; one depends on the current job and the other on the immediately following job. Although, they considered the additive model just to be mathematical, Mason and Anderson [81] and Proust et al. [144] provided a physical definition in which the first component is defined to be setdown and the second to be setup. The setdown operation entails removing the current job from a machine, and the setup operation consists of preparing the machine for the following job. They called this combined operation 'changeover'.

5.3. Sequence-independent batch setups

For batch setup times with group technology assumption, Baker [172] generalized the results of Hitomi and Ham [173] for the two-machine case with separable setups, and provided an optimal solution consisting of two steps. In the first step the order of the jobs within each batch is obtained, and in the second step the batches are sequenced. Extensions of their work to m-stage flowshops are described by Hitomi et al. [174] and Ham et al. [175]. Logendran and Sriskandarajah [9] also addressed the two-machine batch scheduling problem when there is blocking and zero-buffer for the makespan objective. After showing the problem to be NP-hard, they presented a heuristic algorithm and analyzed it in the worst-case performance context under two cases. In the first case the sum of setup times on the second machine is smaller than the sum of processing times on the second machine, whereas in the second case the opposite is true. Vakharia and Chang [176] described a simulated annealing approach to minimize makespan for flowshop problems with batch setup times. Skorin-Kapov and Vakharia [177] developed a tabu search approach to minimize makespan in a flowshop with batch setup times. A genetic algorithm for minimizing makespan, total flow time or both is described by Sridhar and Rajendran [178].

Zdrzalka [179] relaxed the group technology assumption, developed heuristics for the two-machine batch scheduling problem with batch setup times and investigated their worst-case performance. Sotoskov et al. [180] described constructive heuristics for solving the permutation flowshop problems with batch setup times. Danneberg et al. [181] evaluated various heuristics for the flowshop problems with batch setup times and limited buffer availability between machines.

Li [182] extended Gupta and Tunc's [137] problem by grouping parts into part types and batches (families) of part types; thus allowing parts in a part type to share a minor setup and part types within a batch to share a major setup. She compared two allocation policies at stage two and developed six heuristic rules to improve makespan. Unlike Gupta and Tunc [137], Li [182] did not consider removal times.

5.4. Sequence-dependent batch setups

The only known results for the flowshop scheduling problems with sequence dependent batch setup times are those of Vakharia et al. [183] and Schaller et al. [184] who presented branch and bound approaches as well as several heuristics to solve the $F_m|\mathrm{ST_{sd,b}}|C_{\max}$ problem.

6. Job shop problems

A job shop environment consists of different machines and an arriving job may require some or all of the machines in some specific order. A job cannot use the same machine more than once.

Setup considerations in job shops started almost three decades ago when Wilbrecht and Prescott [2] discovered using a simulation study that sequence-dependent setup times play a critical role in the performance of a job shop operating at or near full capacity. While our emphasis in this review has been directed towards the solution of static scheduling problems, research in dynamic scheduling problems have been directed towards the development and evaluation of dispatching rules. For dynamic job shops, Hershauer [185], Flynn [186], and Kim and Bobrowski [22] also used simulation to study the effect of sequence-dependent setup times. Kim and Bobrowski [22] examined the implications of setup times by testing different dispatching rules in environments defined by due date tightness, setup times and cost structure. They found that setup times must be given explicit consideration in solving the scheduling problems especially when they are sequence-dependent.

Again, using simulation, Low [187] compared the performance of his algorithm for sequence-dependent setup times under various criteria against non-sequence-dependent situations, and O'Grady and Harrison [188] proposed a search sequencing rule which prioritizes jobs using a linear combination of the due dates, processing times, and sequence-dependent setup times. For a particular job shop environment, the search sequencing rule performed better than existing rules.

Zhou and Egbelu [189] designed a heuristic algorithm to minimize makespan in a flexible manufacturing system environment with sequence-dependent setup times. The heuristic allows interactions with a human expert via a computer graphic interface. Choi and Korkmaz [190] provided a mixed integer programming formulation for the same problem, and developed a polynomial heuristic that yields better performance than Zhou and Egbelu [189].

Gupta [191] provided a branch-and-bound based algorithm for minimizing total setup cost in a job shop with sequence-dependent setup time. Brucker and Thiele [192] developed a branch and bound algorithm for the general job shop problem with sequence dependent setup times. For minimizing maximum lateness, Ovacik and Uzsoy [71] developed scheduling techniques which use real-time shop floor information. Upon comparison of these techniques with myopic dispatching rules in a job shop environment, semiconductor testing facility and a reentrant flow shop, they found that performance improvements are more significant for shop configurations with constraint resources and reentrant product flow. This is in agreement to the results reported by Patterson [8] who analyzed sequence-dependent setup time with constrained resources.

An open shop is different from a job shop in that a job may be processed in the machines in any sequence the job needs. To our knowledge, the only work that considers setup time in an open shop is by Strusevich [193] who provided an algorithm that has a linear computational time with respect to the number of jobs to solve the two-machine open shop problem for sequence-independent setup and removal times with the makespan criterion.

There are no reported results in job shop scheduling with batch processing times except those of Sotskov et al. [194] who developed constructive heuristics for minimizing several objective functions where batch setup times are sequence independent.

7. Summary and directions for future research

This paper provided a comprehensive review of research directed towards the solution of static scheduling problems involving setup considerations. We classified the literature according to the shop environment, then into batch and non-batch jobs, and finally into sequence-independent and sequence-dependent setup. In a batch problem, setup is incurred when switching between jobs belonging to different batches. For sequence-dependent setup, the setup time of the current operation at a machine depends on both the current and the immediately preceding job at that machine, while in a sequence-independent setup, it only depends on the current job at that machine.

From the review of literature described in this paper and a comparison of the results in Tables 1–3, 5 and 6, some biases of the current research efforts become evident. These observations lead to the following useful directions for future research.

7.1. Emphasize solution of multi-machine scheduling problems

As in all scheduling research, single-machine problems have received the most attention in the setup literature due to their relative simplicity. As seen from Tables 2, 3 and 6, except for single machine cases, scheduling research involving batch setups is still in very early stages. Aside from a few papers on flowshops and parallel-machine shops, research on scheduling with batch setups is mainly in the single-machine environments. Therefore, far more emphasis and research effort are required on solving scheduling problems with batch setup times for the parallel machine and multi-stage scheduling problems. This is particularly true when both major and minor setup times are considered where major setup times may be sequence dependent.

7.2. Emphasize scheduling models with sequence dependent setup times

For parallel machines and multi-stage shops, aside from a couple of papers on sequence-dependent scheduling, the majority of work has been on sequence-independent setup times models. For a flowshop, only a few papers have been found in sequence-dependent scheduling. This shows the need for research on the parallel-machine and multi-stage sequence-dependent scheduling problems.

7.3. Emphasize scheduling models with setup and removal times

Since sequence dependent setup times may be approximated by the sum of sequence independent setup and removal times, research in developing and solving scheduling models which include separable and sequence independent setup and removal times is both interesting and useful. This is true for the single-and-parallel-machine scheduling problems as well as multi-stage scheduling problems.

7.4. Emphasize due date related objectives

Except for single machine problems, the majority of scheduling research considers the makespan as performance measure. Obviously, attention needs to be directed toward other customer-driven performance measures such as the ones related to due-dates. Thus, for example, consideration of minimizing the total weighted earliness and tardiness, weighted number of tardy jobs, and maximum tardiness become important in the development and solution of scheduling models. Thus, future research in the solution of single-machine, parallel-machine and multi-stage scheduling problems with setup times to optimize due date related criteria is essential.

7.5. Emphasize multi-criteria scheduling models

Except for a few cases of single machine problems, practically no research results are available for optimizing multi-criteria scheduling problems in the presence of separable setup times. Since most practical problems involve both setup considerations and multiple objectives, future research in solving scheduling problems with setups to optimize multiple objectives is both desirable and interesting.

7.6. Emphasize generalized shop environments

From the review of literature in scheduling theory summarized in this paper, it is clear that except for very simple cases of hybrid flowshop scheduling problem, no scheduling models have been developed where there are multiple machines at each stage, multiple operations of the same job on the same machine, or time-lags between the multiple operations of the same job. Such generalized environments are quite common in the current and emerging applications of scheduling theory and require explicit consideration of setup times and costs. Development of scheduling models for these environments will certainly help the application of scheduling theory to practical problems.

7.7. Emphasize scheduling models for no-wait processing

The chemical and processing industry applications frequently require continuous processing of jobs through the shop in the presence of major and minor setups. However, only a couple of papers addressed scheduling problems that fall into the combined category of no-wait and separable setup times. Consideration of these aspects in the development and solution procedures for scheduling models provides several challenges for future research.

7.8. Emphasize the theoretical analysis of heuristics

Research on solving scheduling problems involving setup considerations is often devoted to the development of heuristics and their empirical analysis. While such efforts are useful, they do not fully exploit the theoretical developments available for solving scheduling problems without setups. Thus, for example, only in rare cases, some worst-case performance bounds for heuristics to solve scheduling problems with setups are available. Future research efforts to develop a theoretical basis for scheduling models and worst-case analysis of heuristic algorithms will be useful from an academic as well as a practical viewpoint.

7.9. Emphasize solution of stochastic scheduling problems

Stochastic scheduling problems, where some characteristics of the job are modeled as random variables and/or machines may be subject to random breakdowns, with separable setup times have been addressed only in a couple of papers. This is contrary to non-separable setup times research, where many papers have been written. Therefore, another avenue of research is stochastic scheduling problems with setup times.

From the above analysis of the current status of scheduling research involving setup considerations and suggested directions for future research, it is evident that much remains to be done in this area. Therefore, scheduling research involving setup considerations is a fertile field for future research.

Acknowledgements

The research of Ali Allahverdi and Tariq Aldowaisan was supported by Kuwait University Research Administration project number EM-117. The authors are grateful to the three anonymous referees for their useful and constructive comments and suggestions.

References

[1] Ostwald PF. Cost estimating. Salvendy G, editor. Handbook of Industrial Engineering, 2nd ed. 1992, p. 1263–1288.

[2] Wilbrecht JK, Prescott WB. The influence of setup time on job shop performance. Manage Sci 1969;16:B274–B280.

[3] Panwalkar SS, Dudek RA, Smith ML. Sequencing research and the industrial scheduling problem. Elmaghraby SE, editor. Symposium on the Theory of Scheduling and its Applications. 1973, pp. 29–38.

[4] Flynn BB. The effects of setup time on output capacity in cellular manufacturing. Int J Prod Res 1987;25:1761–72.

[5] Wortman DB. Managing capacity: getting the most from your firm's assets. Ind Eng 1992;24:47–9.

[6] Krajewski LJ, King BE, Ritzman LP, Wong DS. Kanban, MRP, and shaping the manufacturing environment. Manage Sci 1987;33:39–57.

[7] Yang J, Deane RH. Setup reduction and competitive advantage in a closed manufacturing cell. Eur J Oper Res 1993;69:413–23.

[8] Patterson MC. Analysis of setup time at constraint resources. Int J Prod Res 1993;31:845–9.

[9] Logendran R, Sriskandarajah C. Two-machine group scheduling problem with blocking and anticipatory setups. Eur J Oper Res 1993;69:467–81.

[10] Unal AT, Uzsoy R, Kiran AS. Rescheduling on a single machine with part-type dependent setup times and deadlines. An Oper Res 1997;70:93–113.

[11] Baker KR. Heuristic procedures for scheduling job families with setups and due dates. Working Paper No. 312, The Amos Tuck School of Business Administration, Hanover, NH 03755: Dartmouth College, 1997.

[12] Bruno J, Downey P. Complexity of task sequencing with deadlines, set-up times and changeover costs. SIAM J Comput 1978;7:393–404.

[13] Monma CL, Potts CN. On the complexity of scheduling with batch setups. Oper Res 1989;37:798–804.

[14] Chen B. A better heuristic for preemptive parallel machine scheduling with batch setup times. SIAM J Comput 1993;22:1303–18.

[15] Pinedo M. Scheduling: Theory, Algorithms, and Systems. Prentice Hall, NJ, 1995.

[16] Conway RW, Maxwell WL, Miller LW. Theory of Scheduling, Addison Wesley, MA, 1967.

[17] Das SR, Gupta JND, Khumawala BM. A saving index heuristic algorithm for flowshop scheduling with sequence dependent set-up times. J Oper Res Soc 1995;46:1365–73.

[18] Franca PM, Gendreau M, Laporte G, Muller FM. A tabu search heuristic for the multiprocessor scheduling problem with sequence dependent setup times. Int J Prod Econ 1996;43:79–89.

[19] Srikar BN, Ghosh S. A MILP model for the *n*-job, *M*-stage flowshop, with sequence dependent setup times. Int J Prod Res 1986;24:1459–72.

[20] Bianco L, Ricciardelli S, Rinaldi G, Sassano A. Scheduling tasks with sequence-dependent processing times. Naval Res Logist 1988;35:177–84.

[21] Bitran GR, Gilbert SM. Sequencing production on parallel machines with two magnitudes of sequence-dependent setup cost. J Manufact Oper Manage 1990;3:24–52.

[22] Kim SC, Bobrowski PM. Impact of sequence-dependent setup times on job shop scheduling performance. Int J Prod Res 1994;32:1503–20.

[23] Ovacik IM, Uzsoy RA. Shifting bottleneck algorithm for scheduling semiconductor testing operations. J Electron Manufact 1992;2:119–34.

[24] Sule DR, Huang KY. Sequency on two and three machines with setup, processing and removal times separated. Int J Prod Res 1983;21:723–32.

[25] Bahl HC, Ritzman LP, Gupta JND. Determining lot sizes and resource requirements: a review. Oper Res 1987;35:329–45.

[26] Missbauer H. Order release and sequence dependent setup times. Int J Prod Econ 1997;49:131–43.

[27] Potts CN, van Wassenhove LN. Integrating scheduling with batching and lot-sizing: a review of algorithms and complexity. J Oper Res Soc 1992;43:395–406.

[28] Vickson RJ, Alfredsson BE. Two and three-machine flowshop scheduling problems with equal size transfer batches. Int J Prod Res 1992;30:1551–75.

[29] Cetinkaya FC, Kayaligil MS. Unit-sized transfer batch scheduling with setup times. Comput Ind Eng 1992;22:177–83.

[30] Cetinkaya FC. Lot streaming in a 2-stage flowshop with setup, processing, and removal times separated. J Oper Res Soc 1994;45:1445–55.

[31] Glass CA, Gupta JND, Potts CN. Lot streaming in 3-stage production processes. Eur J Oper Res 1994;75:378–94.

[32] Vickson RG. Optimal lot streaming for multiple products in a two-machine flowshop. Eur J Oper Res 1995;85:556–75.

[33] Trietsch D, Baker KR. Basic techniques for lot streaming. Oper Res 1993;6:1065–76.

[34] Chen J, Steiner G. Lot streaming with detached setups in 3-machine flowshops. Eur J Oper Res 1997;96:591–611.

[35] Sen T, Gupta SK. A state-of-art survey of static scheduling research involving due dates. OMEGA Int J Manage Sci 1984;12:63–76.

[36] Cheng TCE, Gupta MC. Survey of scheduling research involving due date determination decisions, 1989;38:156–166.

[37] Gupta SK, Kyparisis J. Single-machine scheduling research. OMEGA Int J Manage Sci 1987;15:207–27.

[38] Tang CS. Scheduling batches on parallel machines with major and minor set-ups. Eur J Oper Res 1990;46:28–37.

[39] Lawler EL, Lenstra JK, Rinnooy Kan AHG, Shmoys DB. Sequencing and scheduling: algorithms and complexity. In Logistics of Production and Inventory. Graves SC, Rinnooy Kan AHG, Zipkin PH, editors. Amsterdam, Netherlands: North Holland, 1993, p. 445–522.

[40] Glassey CR. Minimum changeover scheduling of several products on one machine. Oper Res 1968;16:342–52.
[41] Mitsumori S. Optimal production scheduling of multicommodity in flow line. IEEE Trans Syst Man Cybern 1972;2:486–93.
[42] Miyazaki S, Ohta H. Backward scheduling to minimize the actual mean flow time with dependent and independent setup times. Proc. IXth ICPR, 1987, p. 2680–2686.
[43] Zdrzalka S. Preemptive scheduling with release dates, delivery times and sequence independent setup times. Eur J Oper Res 1994;76:60–71.
[44] Laporte G. The traveling salesman problem: An overview of exact and approximate algorithms. Eur J Oper Res 1992;59:231–47.
[45] Junger M, Reinelt G, Rinaldi G. The traveling salesman problem. in Annotated bibliographies in combinatorial optimization. Dell'Amico M, Maffioli F, Marcello S, editors. Wiley, 1997.
[46] Gilmore PC, Gomory RE. Sequencing a one-state variable machine; a solvable case of the traveling salesman problem. Oper Res 1964;12:655–79.
[47] Presby JT, Wolfson ML. An algorithm for solving job sequencing problems. Manage Sci 1967;13:B454–B464.
[48] Driscoll WC, Emmons H. Scheduling production on one machine with changeover costs. AIIE Trans 1977;9:388–95.
[49] Burstall RM. A heuristic method for a job-shop scheduling problem. Oper Res Q 1966;17:291–304.
[50] Buzacott JA, Dutta SK. Sequencing many jobs on a multi-purpose facility. Naval Res Logist Q 1971;18:75–82.
[51] Taha H. Sequencing by implicit ranking and zero-one polynomial programming. AIIE Trans 1971;3:299–301.
[52] Sielken RL, Jr. Sequencing with set-up costs by zero-one mixed integer linear programming. AIIE Trans 1976;8:369–71.
[53] Barnes JW, Vanston LK. Scheduling jobs with linear delay penalties and sequence dependent setup costs. Oper Res 1981;29:146–60.
[54] Kusiak A, Finke G. Modeling and solving the flexible forging module scheduling problem. Eng Optim 1987;12:1–12.
[55] Laguna M, Barnes JW, Glover FW. Tabu search methods for a single-machine scheduling problem. J Intell Manufact 1991;2:63–74.
[56] Laguna M, Glover FW. Integrating target analysis and tabu search for improved scheduling systems. Exp Sys Appl 1993;6:287–97.
[57] Feo TA, Sarathy K, McGahan J. A grasp for single machine scheduling with sequence dependent setup costs and linear delay penalties. Comput Oper Res 1996;23:881–95.
[58] Kolahan F, Liang M, Zuo M. Solving the combined part sequencing and tool replacement problem for an automated machine center: A Tabu search approach. Comput Ind Eng 1995;28:731–43.
[59] Gavett JW. Three heuristic rules for sequencing jobs to a single production facility. Manage Sci 1965;11:B166–B176.
[60] Haynes RD, Komar CA, Byrd J. The effectiveness of three heuristic rules for job sequencing in a single production facility. Manage Sci 1973;9:575–80.
[61] White CH, Wilson RC. Sequence dependent setup times and job sequencing. Int J Prod Res 1977;15:191–202.
[62] Woodruff DL, Spearman ML. Sequencing and batching for two classes of job with deadlines and setup times. Prod Oper Manage 1992;1:87–102.
[63] Farn CD, Muhlemann AP. The dynamic aspects of a production scheduling problem. Int J Prod Res 1979;17:15–21.
[64] Lockett AG, Muhlemann AP. A scheduling problem involving sequence dependent changeover costs. Oper Res 1972;20:895–902.
[65] Moore JE. An algorithm for a single-machine scheduling problem with sequence-dependent setup times and scheduling windows. AIIE Trans 1975;7:35–41.
[66] Arcelus FJ, Chandra R. On n–1–F setup dependent problems. Eng Opt 1983;7:59–67.
[67] He W, Kusiak A. Scheduling manufacturing systems. Comput Ind 1992;20:163–75.
[68] Picard JC, Queyranne M. The time-dependent traveling salesman problem and its application to the tardiness problem in one machine scheduling. Oper Res 1978;26:86–110.
[69] Uzsoy R, Martin-Vega LA, Lee CY, Leonard PA. Production scheduling algorithms for a semiconductor test facility. IEEE Trans Semicond Manufact 1991;4:270–80.
[70] Uzsoy R, Lee C, Martin-Vega A. Scheduling semiconductor test operations: Minimizing maximum lateness and number of tardy jobs on a single machine. Naval Res Logist 1992;39:369–88.
[71] Ovacik IM, Uzsoy R. Exploiting shop floor status information to schedule complex job shops. J Manufact Syst 1994;13:73–84.
[72] Ovacik IM, Uzsoy R. Rolling horizon algorithms for a single-machine dynamic scheduling problem with sequence-dependent setup times. Int J Prod Res 1994;32:1243–63.
[73] Rubin PA, Ragatz GL. Scheduling in a sequence dependent setup environment with genetic search. Comput Oper Res 1995;22:85–99.
[74] Kim S-Y, Lee Y-H, Agnihotri D. A hybrid approach to sequencing jobs using heuristic rules and neural networks. Prod Planning Control 1995;6:445–54.
[75] Lee YH, Bhaskaran K, Pinedo M. A heuristic to minimize the total weighted tardiness with sequence dependent setups. IIE Trans 1997;29:45–52.
[76] Raman N, Rachamadugu RV, Talbot FB. Real-time scheduling on an automated machine center. Eur J Oper Res 1989;40:222–42.
[77] Asano M, Ohta H. Single machine scheduling using dominance relation to minimize earliness subject to ready and due times. Int J Prod Econ 1996;44:35–43.
[78] ten Kate HA, Wijngaard J, Zijm WHM. Minimizing total earliness, tardiness, and setup costs. Research Report No. RR 1991-12, School of Management and Organization, University of Groningen, Holland, 1991.

[79] Tan KC, Narasimhan R. Multi-objective sequencing with sequence dependent setup times. Int J Oper Quant Manage 1997;3:69–84.
[80] Bruno J, Sethi R. Tasks sequencing in a batch environment with setup times. Found Control Eng 1978;3:105–17.
[81] Mason AJ, Anderson EJ. Minimizing flow time on a single machine with job classes and setup times. Naval Res Logist 1991;38:333–50.
[82] Herrmann JW, Lee C-Y. Solving a class scheduling problem with a genetic algorithm. ORSA J Comput 1995;7:443–52.
[83] Mason AJ. Genetic algorithms and scheduling problems. PhD thesis, Department of Management Sciences, University of Cambridge, UK, 1992.
[84] Webster S, Baker KR. Scheduling groups of jobs on a single machine. Oper Res 1995;43:692–703.
[85] Ozden M, Egbelu PJ, Iyer A. Job scheduling in a group technology environment for a single facility. Comput Ind Eng 1985;9:67–72.
[86] Liao C-J, Liao L-M. Single machine scheduling with major and minor setups. Comp Oper Res 1997;24:169–78.
[87] Gupta JND, Ho JC, van der Veen JAA. Single machine hierarchical scheduling with customer orders and multiple job classes. An Oper Res 1997;70:127–43.
[88] Liao C-J. Tradeoffs between setup times and carrying costs for finished items. Comp Oper Res 1993;20:697–705.
[89] Zdrzalka S. Approximation algorithm for single-machine sequencing with delivery times and unit batch set-up times. Eur J Oper Res 1991;51:199–209.
[90] Zdrzalka S. Analysis of approximation algorithms for single-machine scheduling with delivery times and sequence independent batch setup times. J Oper Res Soc 1995;80:371–80.
[91] Baker KR, Magazine MJ. Minimizing maximum lateness with job families Working Paper No. 314, The Amos Tuck School of Business Administration, Hanover, NH 03755: Dartmouth College, 1997.
[92] Ghosh JB, Gupta JND. Batch scheduling to minimize maximum lateness. Oper Res Lett, to appear.
[93] Constantopuolos P, Georgiou T, Tsatsakis N. One-machine scheduling with setup times; fast solutions and applications. Advances in Production Management Systems, B-13. Pappas IA, Tatsiopoulos IP, editors. North-Holland: Elsevier Science Publishers B.V., 1993.
[94] Williams D, Wirth A. A new heuristic for a single machine problem with setup times. J Oper Res Soc 1996;47:175–80.
[95] Crauwels AJ, Potts CN, van Wassenhove LN. Local search heuristics for single machine scheduling with batch set-up times to minimize total weighted completion time. An Oper Res 1997;70:261–79.
[96] Schutten JMJ, van de Velde SL, Zijm WHM. Single-machine scheduling with release dates, due dates and family setup times. Manage Sci 1996;42:1165–74.
[97] Hariri AMA, Potts CN. Single machine scheduling with batch set-up times to minimize maximum lateness. An Oper Res 1997;70:75–92.
[98] Unal AT, Kiran AS. Batch sequencing. IIE Trans Sched Logist 1992;24:73–83.
[99] Sahney VK. Single-server, two-machine sequencing with switching time. Oper Res 1972;20:24–36.
[100] Gupta JND. Optimal schedules for single facility with two job classes. Comp Oper Res 1984;11:409–13.
[101] Potts CN. Scheduling two job classes on a single machine. Comp Oper Res 1991;18:411–5.
[102] Gupta JND. Single facility scheduling with multiple job classes. Eur J Oper Res 1988;8:42–5.
[103] Psaraftis HN. A dynamic programming approach for sequencing groups of identical jobs. Oper Res 1980;28:1347–59.
[104] Ozgur C, Brown JR. A two-stage traveling salesman procedure for the single machine sequence-dependent scheduling problem. OMEGA 1995;23:205–19.
[105] Sawicki JD. The problems of tardiness and saturation in a multi class queue with sequence-dependent setups. AIIE Trans 1973;5:250–5.
[106] Ahn BH, Hyun JH. Single facility multi-class job scheduling. Comp Oper Res 1990;17:265–72.
[107] Ghosh JB. Batch scheduling to minimize total completion time. Oper Res Lett 1994;16:271–5.
[108] Chen Z-L. Scheduling with batch setup times and earliness–tardiness penalties. Eur J Oper Res 1997;96:518–37.
[109] Tang CS, Wittrock RJ. Parallel machine scheduling with major and minor setups. York Town Heights, NY: RC 11412, IBM TJ Watson Research Center, 1985.
[110] Wittrock RJ. Scheduling parallel machines with setups. York Town Heights, NY: RC 11740, IBM TJ Watson Research Center, 1986.
[111] Cheng TCE, Chen Z-L. Parallel machine scheduling with batch setup times. Oper Res 1994;42:1171–4.
[112] Monma CL, Potts CN. Analysis of heuristics for preemptive parallel machine scheduling with batch setup times. Oper Res 1993;41:981–93.
[113] So KC. Some heuristics for scheduling jobs on parallel machines with setups. Manage Sci 1990;36:467–75.
[114] Wittrock RJ. Scheduling parallel machines with major and minor setups. Int J Flex Manufact Syst 1990;2:329–41.
[115] Dietrich BL. A Two-phase heuristic for scheduling on parallel unrelated machines with setups. York Town Heights, NY: RC 14330, IBM TJ Watson Research Center, 1989.
[116] Schutten JMJ, Leussink RAM. Parallel machine scheduling with release dates, due dates and family setup times. Int J Prod Econ 1996;46-47:119–25.
[117] Brucker P, Kovalyov MY, Shafransky YM, Werner F. Batch scheduling with deadlines and parallel machines. An Oper Res, to appear.
[118] Marsh JD, Montgomery DC. Optimal procedures for scheduling jobs with sequence-dependent changeover times on parallel processors. AIIE Technical Papers, 1973, p. 279–286.
[119] Deane RH, White ER. Balancing workloads and minimizing setup costs in the parallel processing shop. Oper Res Q 1975;26:45–53.
[120] Geoffrion AM, Graves GW. Scheduling parallel production lines with changeover costs: Practical appli-

cation of a quadratic assignment/LP approach. Oper Res 1976;24:595–610.
[121] Guinet A. Textile production systems: a succession of non-identical parallel processor shops. J Oper Res Soc 1990;42:655–71.
[122] Elmaghraby SE, Guinet A, Schellenberger KW. Sequencing on parallel processors: an alternate approach. OR Technical Report No. 273, Raleigh, NC: North Carolina State University, 1993.
[123] Parker RG, Deane RH, Holmes RA. On the use of a vehicle routing algorithm for the parallel processor problems with sequence dependent changeover costs. AIIE Trans 1977;9:155–60.
[124] Dearing PM, Henderson RA. Assigning looms in a textile weaving operation with changeover limitations. Prod Inv Manage 1984;25:23–31.
[125] Sumichrast R, Baker JR. Scheduling parallel processors: an integer linear programming based heuristic for minimizing setup time. Int J Prod Res 1987;25:761–71.
[126] Ovacik IM, Uzsoy R. Worst-case error bounds for parallel machine scheduling problems with bounded sequence-dependent setup times. Oper Res Lett 1993;14:251–6.
[127] Ovacik IM, Uzsoy R. Rolling horizon procedures for dynamic parallel machine scheduling with sequence-dependent setup times. Int J Prod Res 1995;33:3173–92.
[128] Guinet A, Dussauchoy A. Scheduling sequence dependent jobs on identical parallel machines to minimize completion time criteria. Int J Prod Res 1993;31:1579–94.
[129] Lee YH, Pinedo M. Scheduling jobs on parallel machines with sequence dependent setup times. Eur J Oper Res 1997;100:464–74.
[130] Johnson SM. Optimal two- and three-stage production schedules with setup times included. Naval Res Logist Q 1954;1:61–8.
[131] Corwin BD, Esogbue AO. Two-machine flowshop scheduling problems with sequence dependent setup times: a dynamic programming approach. Naval Res Logist Q 1974;21:515–24.
[132] Yoshida T, Hitomi K. Optimal two-stage production scheduling with setup times separated. AIIE Trans 1979;11:261–3.
[133] Gupta JND, Darrow WP. Approximate schedules for the two-machine flowshop with sequence dependent setup times. Indian J Manage Syst 1985;1:6–11.
[134] Gupta JND. Optimal scheduling in multistage flowshop. AIIE Trans 1972;4:238–43.
[135] Dell'Amico M. Shop problems with two machines and time lags. Oper Res 1996;44:777–87.
[136] Strusevich VA, Zwaneveld C. On non-permutation solutions to some two-machine flow shop scheduling models. Z Oper Res 1994;39:305–19.
[137] Gupta JND, Tunc EA. Scheduling a two-stage hybrid flowshop with separable setup and removal times. Eur J Oper Res 1994;77:415–28.
[138] Sule DR. Sequencing n jobs on two machines with setup, processing and removal times separated. Naval Res Logist Q 1982;29:517–9.
[139] Hall NG, Sriskandarajah C. A survey of machine scheduling problems with blocking and no-wait in process. Oper Res 1996;44:510–25.
[140] Aldowaisan T, Allahverdi A. Total flowtime in no-wait flowshops with separated setup times. Comput Oper Res, 1988;25:757–65.
[141] Khurana K, Bagga PC. Scheduling of job-block with deadline in $n \times 2$ flowshop problem with separated setup times. Indian J Pure Appl Math 1985;16:213–24.
[142] Allahverdi A. Two-stage production scheduling with separated set-up times and stochastic breakdowns. J Oper Res Soc 1995;46:896–904.
[143] Allahverdi A. Scheduling in stochastic flowshops with independent setup, processing and removal times. Comput Oper Res 1997;24:955–60.
[144] Proust C, Gupta JND, Deschamps V. Flowshop scheduling with set-up, processing and removal times separated. Int J Prod Res 1991;29:479–93.
[145] Rajendran C, Ziegler H. Heuristics for scheduling in a flowshop with setup, processing, and removal times separated. Prod Planning Control 1997;8:568–76.
[146] Khurana K, Bagga PC. Minimizing the makespan in a two-machine flowshop with time lags and setup conditions. Z Oper Res 1984;28:163–74.
[147] Szwarc W. The flowshop problem with time lags and separated setup times. ZOR-Z Oper Res 1986;30:B15–B22.
[148] Szwarc W. Flowshop problems with time-lags. Manage Sci 1983;29:477–81.
[149] Nabeshima I, Maruyama S. Two- and three-machine flowshop makespan scheduling problems with additional times separated from processing times. J Oper Res Soc Jpn 1984;27:348–66.
[150] Dileepan P, Sen T. Job lateness in a two-machine flowshop with setup times separated. Comput Oper Res 1991;18:549–56.
[151] Allahverdi A, Aldowaisan T. Job lateness in flowshops with setup and removal times separated. J Oper Res Soc, to appear.
[152] Allahverdi A. Minimizing mean flowtime in a two-machine flowshop with sequence independent setup times. Comput Oper Res, revised.
[153] Gupta JND, Strusevich VA, Zwaneveld C. Two-stage no-wait scheduling models with setup and removal times. Comput Oper Res 1997;24:1025–31.
[154] Cao J, Bedworth DC. Flow shop scheduling in serial multi-product processes with transfer and set-up times. Int J Prod Res 1992;30:1819–30.
[155] Park T, Steudel HJ. Analysis of heuristics for job sequencing in manufacturing flow line work-cells. Comput Ind Eng 1991;20:129–40.
[156] Uskup G, Smith SB. A branch and bound for two-stage production sequencing. Oper Res 1975;23:118–36.
[157] Gupta JND, Darrow WP. The two-machine sequence dependent flowshop scheduling problem. Eur J Oper Res 1986;24:439–46.
[158] Gupta JND. Flowshop schedules with sequence dependent setup times. J Oper Res Soc Jpn 1986;29:206–19.
[159] Gupta JND. A search algorithm for the generalized scheduling problem. Comput Oper Res 1975;2:83–90.

[160] Stafford EF, Tseng FT. On the Srikar–Ghosh MILP model for the NxM SDST flowshop problem. Int J Prod Res 1990;28:1817–30.

[161] Rios-Mercado RZ, Bard FB. The flowshop scheduling polyhedron with setup times. Technical Report ORP96-07, Graduate Program in Operations Research. Austin, TX 78712-1063: University of Texas at Austin, 1996.

[162] Gupta JND, Das SR, Ghosh S. Flowshop Scheduling with Sequence Dependent Setup Times. Working paper, Department of Management. Muncie, IN 47306: Ball State University, 1995.

[163] Rios-Mercado RZ, Bard FB. Computational experience with a branch-and-cut algorithm for flowshop scheduling with setups. Comput Oper Res, 1988;25:351–66.

[164] Gupta JND. Flowshop Scheduling with Sequence Dependent Setup Times. Proceedings of the ORSA/TIMS national meeting, Washington, DC, April 1988.

[165] Simons JV. Heuristics in flow shop scheduling with sequence dependent setup times. Omega 1992;20:215–25.

[166] Rios-Mercado RZ, Bard FB. An enhanced TSP-based heuristic for makespan minimization in a flowshop with setup times. Technical Report ORP97-08, Graduate Program in Operations Research, Austin, TX 78712-1063: University of Texas at Austin, (1997).

[167] Rios-Mercado RZ, Bard FB. Heuristics for the flow line problem with setup costs. Eur J Oper Res, to appear.

[168] Parthasarathy S, Rajendran C. A simulated annealing heuristic for scheduling to minimize weighted tardiness in a flowshop with sequence dependent setup times of jobs – a case study. Prod Planning Control 1997;8:475–83.

[169] Parthasarathy S, Rajendran C. An experimental evaluation of heuristics for scheduling in a real-life flowshop with sequence dependent setup times of jobs. Int J Prod Econ 1997;49:255–63.

[170] Rajagopalan D, Karimi IA. Scheduling in serial mixed-storage multiproduct processes with transfer and set-up times. Foundations of Computer Aided Process Operations, Reklaitis GV, Spriggs HD, editors. New York: Elsevier Science Publishers, 1987, p. 679–686.

[171] Szwarc W, Gupta JND. A flow-shop problem with sequence-dependent additive setup times. Naval Res Logist 1987;34:619–27.

[172] Baker KR. Scheduling groups of jobs in the two-machine flowshop. Math Comput Model 1990;13:29–36.

[173] Hitomi K, Ham I. Operations scheduling for group technology applications. CIRP An 1976;25:419–22.

[174] Hitomi K, Nakamura N, Yoshida T, Okuda K. An Experimental Investigation of Group Production Scheduling. Proceedings of the 4th International Conference on Production Research, Tokyo, 1977, p. 608–617.

[175] Ham I, Hitomi K, Yoshida T. Group Technology. Boston: Kluwer-Nijhoff Publishing, 1985.

[176] Vakharia AJ, Chang Y-L. A simulated annealing approach to scheduling a manufacturing cell. Naval Res Logist 1990;37:559–77.

[177] Skorin-Kapov J, Vakharia AJ. Scheduling a flow-line manufacturing cell: a tabu search approach. Int J Prod Res 1993;31:1721–34.

[178] Sridhar J, Rajendran C. A genetic algorithm for family and job scheduling a flow-line based manufacturing cell. Comput Ind Eng 1994;27:469–72.

[179] Zdrzalka S. Two-machine flow shop scheduling problem with family setup times. Report no. 16/95, Institute of Engineering Cybernetics, Technical University of Wroclaw, Poland, 1995.

[180] Sotoskov YN, Tautenhahn T, Werner F. Heuristics for permutation flowshop scheduling with batch setup times. OR Spektrum 1996;18:67–80.

[181] Danneberg D, Tautenhahn T, Werner F. A comparison of heuristic algorithms for flow shop scheduling problems with setup times and limited batch size, Preprint. Magdeburg, Germany: Otto-von-Guericke University, 1998.

[182] Li S. A hybrid two-stage flowshop with part family, batch production, major and minor set-ups. Eur J Oper Res 1997;102:142–56.

[183] Vakharia AJ, Schaller JE, Gupta JND. Designing and scheduling manufacturing cells. Proceedings of the INFORMS national meeting, New Orleans, LA, October 1995.

[184] Schaller JE, Gupta JND, Vakharia AJ. Group scheduling with sequence dependent setups. Proceedings of the Annual Decision Sciences Institute Meeting, San Diego, CA, November 1997, p. 1141–1143.

[185] Hershauer JC. An empirically-derived simulation to explore sequencing decisions. Ph.D. Thesis, Indiana University, 1970.

[186] Flynn BB. Group technology vs. process layout: a comparison using computerized job shop simulation. Ph.D. Thesis, Indiana University, 1984.

[187] Low CY. Job shop scheduling heuristics for sequence-dependent setups. Comput Ind Eng 1995;29:279–83.

[188] O'Grady PJ, Harrison C. Search based job scheduling and sequencing with setup times. Omega Int J Manage Sci 1988;16:547–52.

[189] Zhou C, Egbelu PJ. Scheduling in a manufacturing shop with sequence-dependent setups. Robotics Comput-Integr Manufact 1989;5:73–81.

[190] Choi I-C, Korkmaz O. Job shop scheduling with separable sequence-dependent setups. An Oper Res 1997;70:155–70.

[191] Gupta SK. n jobs and m machines job-shop problems with sequence dependent setup times. Int J Prod Res 1982;20:643–56.

[192] Brucker P, Thiele O. A branch and bound algorithm for the general job shop with sequence dependent setup times. OR Spektrum, 1996;18.

[193] Strusevich VA. Two machine open shop scheduling problem with setup, processing and removal times separated. Comput Oper Res 1993;20:597–611.

[194] Sotoskov YN, Tautenhahn T, Werner F. On the application of insertion techniques for job shop problems with batch setup times. RAIRO Oper Res, 1998, to appear.