A Survey of Scheduling Problems with Setup Times or Costs

Ali Allahverdi
Department of Industrial and Management Systems Engineering
Kuwait University, P.O. Box 5969, Safat, Kuwait
Fax: 965 481 6137, e-mail: allahverdi@kuc01.kuniv.edu.kw

C.T. Ng and T.C.E. Cheng
Department of Logistics
The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
e-mail: lgtctng@polyu.edu.hk, lgtcheng@polyu.edu.hk

Mikhail Y. Kovalyov
Faculty of Economics, Belarusian State University and United Institute of Informatics Problems,
National Academy of Sciences of Belarus, Surganova 6, 220012 Minsk, Belarus
e-mail: koval@newman.bas-net.by

Abstract

The first comprehensive survey paper on scheduling problems with separate setup times or costs was conducted by Allahverdi et al. (1999), who reviewed the literature since the mid-1960s. Since the appearance of that survey paper, there has been an increasing interest in scheduling problems with setup times (costs) with an average of more than 40 papers per year being added to the literature. The objective of this paper is to provide an extensive review of the scheduling literature on models with setup times (costs) from then to date covering more than 300 papers. Given that so many papers have appeared in a short time, there are cases where different researchers addressed the same problem independently, and sometimes by using even the same technique, e.g., genetic algorithm. Throughout the paper we identify such areas where independently developed techniques need to be compared. The paper classifies scheduling problems into those with batching and non-batching considerations, and with sequence-independent and sequence-dependent setup times. It further categorizes the literature according to shop environments, including single-machine, parallel machines, flow shop, no-wait flow shop, flexible flow shop, job shop, open shop, and others.

Key words - scheduling, setup time, setup cost, survey (review), single machine, parallel machines, flow shop, job shop, open shop

1. Introduction

The first systematic approach to scheduling problems was undertaken in the mid-1950s. Since then, thousands of papers on different scheduling problems have appeared in the literature. The majority of these papers assumed that the setup time (cost) is negligible or part of the job processing time (cost). While this assumption simplifies the analysis and/or reflects certain applications, it adversely affects the solution quality of many applications of scheduling that require an explicit treatment of setup times (costs).

The interest in scheduling problems that treat setup times (costs) as separate began in the mid-1960s. The corresponding results have been summarized in the survey papers of Allahverdi et al. (1999), Yang and Liao (1999), Cheng et al. (2000), and Potts and Kovalyov (2000). Yang and Liao (1999) concentrated on static and deterministic scheduling problems. Cheng et al. (2000) reviewed flow shop scheduling problems, while Potts and Kovalyov (2000) surveyed scheduling problems with batching. Allahverdi et al. (1999) provided a review of the literature covering dynamic and stochastic problem settings in different shop environments, including single-machine, parallel machines, flow shops, and job shops.

There has been an increase in interest in scheduling problems involving setup times (costs) since the publication of the above surveys, whereby an average of more than 40 papers per year have been added to the literature. The objective of this paper is to review the literature on separate setup times (costs) involving static, dynamic, deterministic, and stochastic problems for all shop environments, including single-machine, parallel machines, flow shops (regular flow shop, no-wait flow shop, flexible flow shop, assembly flow shop), job shops, open shops, and others. The current paper is a continuation of the earlier survey papers of Allahverdi et al. (1999) and Potts and Kovalyov (2000) covering more than 300 papers that were published in 1999-2006.

We do not cite in this survey paper the earlier research that was covered by Allahverdi et al. (1999) and Potts and Kovalyov (2000) even when a comparison of a new result with a result that was referenced in these two papers is required. Therefore, rather than stating that, e.g., Park et al. (2000) proposed the use of a neural network to get values for parameters in calculating a priority rule for the $P/ST_{sd}/\sum w_iT_i$ problem, where their computational results indicated that their proposed approach outperforms that of Lee et al. (1997), we state that Park et al.'s computational results indicated that their proposed approach outperforms that of an earlier approach. This is because Lee et al. (1997) had already been cited in Allahverdi et al. (1999**)** and the already long reference list of the current paper.

We do not review setup time or setup cost research for lot-sizing and scheduling problems in the context of inventory management, see the surveys of Drexl and Kimms (1997) and Karimi et al. (2003), for vehicle routing and scheduling problems, see the review of Laporte (1992), for lot streaming problems with continuous batch sizes, e.g., Chiu and Chang 2005, and Kalir and Sarin 2003, and research on complex industrial problems involving time indexed variables and continuous batch sizes, e.g., Berning et al. 2002.

The importance and applications of scheduling models with explicit considerations of setup times (costs) have been discussed in several studies since the mid-1960s. Following are some recent applications:

- Laguna (1999) considered a facility that produces supplies to photocopiers and laser printers. He pointed out that changing production from one toner to another results in large setup times (generally of the order of days).
- Schaller et al. (2000) addressed the problem of manufacturing printed circuit boards on an automated insertion machine. They stated that the problem is a setup time scheduling problem.
- In the textile industry, setup times are significant and have to be considered as separate as mentioned by Gendreau et al. (2001). Fabric types are assigned to looms equipped with wrap chains. When the fabric type is changed on a machine, the wrap chain must be replaced and the time it takes depends on the current and the previous fabric types.
- Simons and Russel (2002) presented a case study of batching in mass service operations in the example of a court. The interviewed judges noted that setup times and costs are due to trips to court, pre-court meetings, mental preparation, and communication of instructions.
- Many WWW applications require access, transfer, and synchronization of large multimedia data objects (MDOs), such as audio, video, and images, across a communication network. The processing and transfer of large MDOs across the Web affects the response time to end users. Therefore, the MDO scheduling process is a critical aspect of distributed multimedia database systems and it is very important to provide distributed systems with an efficient multimedia data scheduling strategy. Allahverdi and Al-Anzi (2002) showed that the MDO scheduling problem for WWW applications can be modeled as a two-machine flow shop scheduling problem with separate setup times.
- Kim et al. (2002) considered the production of compound semiconductors that are used for electronic components in information displays, mobile telecommunications, and wireless data communications. They pointed out that the machines used in the production of compound semiconductors should be adjusted whenever different types of wafers are diced. Therefore, different setup times are required depending on wafer sequences.
- Chang et al. (2003) described a biaxially oriented polypropylene (BOPP) film factory, which produces products such as adhesive tapes, photo albums, foodstuff packages, book covers, etc. They stated that the time, raw materials, and equipment necessary to prepare for the next job in the factory depend on the preceding job, and therefore, the setup times and setup costs are sequence-dependent.
- Yi and Wang (2003) considered stamping plants that are used by most auto-makers. In such plants, the setup time between manufacturing parts involves the changing of heavy dies, which indicates the significance of setup times.

- Lin and Liao (2003) described a label sticker manufacturing company. They stated that the problem is a two-stage hybrid flow shop, where the first stage is a single high speed machine that is used to glue the surface material and liner together to produce the label stickers. They stated that when the machine in the first stage is changed over from jobs in one class to jobs in another class, a sequence-dependent setup time is required for the changeover task.
- Andrés et al. (2005a) addressed the problem of product grouping in the tile industry and stated that the problem can be modeled as a three-stage hybrid flow shop with separate and sequence-dependent setup times. They pointed out that the objective for such a problem is to minimize the changeover (setup) time in order to reduce the production time.

2. Notation and Classification

This section provides the necessary notation and classification for the scheduling problems with setup times/costs discussed in this paper. The definitions of batch and non-batch setup times (costs) are first introduced.

A batch setup time (cost) occurs when jobs, e.g., machine parts, are processed in *batches* (pallets, containers, boxes) and a setup of a certain time or cost precedes the processing of each batch. The definition of a batch is as follows. The jobs are supposed to be partitioned into F, $F \geq 1$, *families*. A batch is a set of jobs of the same family. While families are supposed to be given in advance, batch formation is a part of the decision making process.

An important special case appears when the *Group Technology* assumption has to be observed. According to this assumption, no family can be split, i.e., only a single batch can be formed for each family.

The batch setup time (cost) can be machine dependent or sequence (of families) dependent. It is sequence-dependent if its duration (cost) depends on the families of both the current and the immediately preceding batches, and is sequence-independent if its duration (cost) depends solely on the family of the current batch to be processed.

Batch setup models are further partitioned into batch availability and job availability models. According to the batch availability model, all the jobs of the same batch become available for processing and leave the machine together. In the job availability model, each job's start and completion times are independent of other jobs in its batch. We implicitly assume that the job availability model is considered, if it is not stated otherwise.

For multistage processing systems, permutation and non-permutation schedules and schedules with consistent and inconsistent batches are distinguished. A schedule is a permutation schedule if the job sequences are the same on all the machines. The batches are consistent if batch formation is the same on all the machines. Opposite statements define a non-permutation schedule and inconsistent batches.

In a non-batch processing environment, a setup time (cost) is incurred prior to the processing of each job. The corresponding model can also be viewed as a batch setup time (cost) model in which each family consists of a single job.

We distinguish anticipatory or non-anticipatory setups. A setup is anticipatory if it can be started before the corresponding job or batch becomes available on the machine. Otherwise, a setup is non-anticipatory. If it is not stated explicitly that setups are non-anticipatory, we assume that they are anticipatory unless there are job release dates, in which case a setup cannot start before the corresponding release date. In setup time models, no job processing is possible on a machine while a setup is being performed on the machine.

We use a similar classification of setup time (cost) problems adopted in the survey paper by Allahverdi et al. (1999). The terminology "family" adopted by Potts and Kovalyov (2000) is used in the current survey to denote initial job partitioning, while the terminology "batch" is used to denote a part of the solution. It should be noted that many publications use the terminology "batch" to denote the initial job partitioning and they use different names like sub-batch, lot, sub-lot, etc., to denote a set of jobs of the same family processed consecutively on the same machine. This terminology was adopted by Allahverdi et al. (1999).

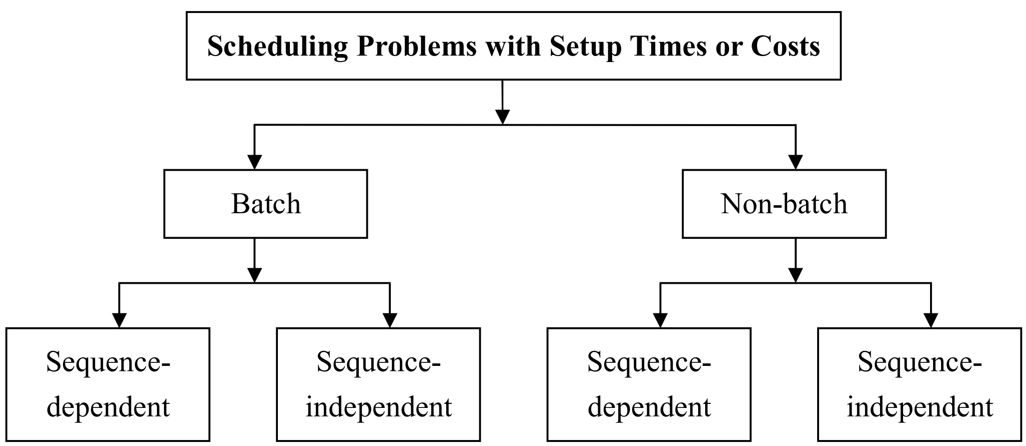

Fig. 1. Classification of separate setup time (cost) scheduling problems.

We adapt the three-field notation $\alpha/\beta/\gamma$ of Graham et al. (1979) to describe a scheduling problem. The α field describes the shop (machine) environment. The β field describes the setup information, other shop conditions, and details of the processing characteristics, which may contain multiple entries. Finally, the γ field contains the objective to be minimized. For example, a three-machine flow shop scheduling problem to minimize maximum lateness with batch sequence-dependent setup times will be noted as $F3/ST_{sd,b}/L_{max}$.

<u>Shop type (α field)</u>

1 Single machine

F	Flow shop
FF	Flexible (hybrid) flow shop
AF	Assembly flow shop
P, Q, R	Parallel machines (P: identical machines; Q: uniform machines; R: unrelated machines)
J	Job shop
O	Open shop

Shop characteristics (β field)

prec	precedence constraints
r_j	non-zero release date
pmtn	preemption

Setup information (β field)

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

Performance criteria (γ field)

C_{max}	makespan
L_{max}	maximum lateness
T_{max}	maximum tardiness
D_{max}	maximum delivery time
TSC	total setup/changeover cost
TST	total setup/changeover time
$\sum f_j$	total flowtime
$\sum C_j$	total completion time
$\sum E_j$	total earliness
$\sum T_j$	total tardiness
$\sum U_j$	number of tardy (late) jobs

Σw_jC_j total weighted completion time
Σw_jU_j weighted number of tardy jobs
Σw_jE_j total weighted earliness
Σw_jT_j total weighted tardiness
Σw_jf_j total weighted flowtime

It should be noted that f_j=C_j if all the jobs are ready at time zero. Also, since minimizing the total or the mean of an objective function results in the same solution, we do not distinguish between the two. For example, the total completion time (ΣC_j) and the mean completion time ($n^{-1}\Sigma C_j$) are equivalent criteria, and therefore, for simplicity we just refer both by ΣC_j.

In some cases, the setup operation is performed by a single server. This problem is denoted by adding a letter *S* after the shop environment. For example, a three-machine parallel scheduling problem to minimize the makespan with a single server for the setup operations is denoted by $P3,S//C_{max}$.

3. Single Machine

This section describes the setup time (cost) literature in a single machine environment, a summary of which is provided in Tables 1 and 2.

3.1. Non-batch sequence-independent setup times

Graves and Lee (1999) addressed the problems $1/ST_{si}/L_{max}$ and $1/ST_{si}/\Sigma w_jC_j$ where machine maintenance must be performed within certain intervals. They assumed that if a job is not finished before the next maintenance starts, then an additional setup is necessary when the remaining processing starts. They showed that both problems are NP-hard when the planning horizon is long, and they proposed pseudo-polynomial time dynamic programming algorithms for each problem.

Liu and Cheng (2002) considered the $1/ST_{si}, pmtn, r_j/D_{max}$ problem where it was assumed that a certain setup time is incurred, which is job dependent, whenever a preempted job is restarted. They proved that the problem is strongly NP-hard (even if each setup time is one unit), and presented a dynamic programming algorithm to solve the problem. The dynamic programming has a pseudo-polynomial time requirement if the number of release dates is constant. Liu and Cheng (2004) addressed the same problem but with the objective function of total weighted completion time, i.e., $1/ST_{si}, pmtn, r_j/\Sigma w_jC_j$, and with the assumption that setup times are constant and job independent. They proved that this problem is also strongly NP-hard, and proposed a greedy algorithm. They too proved that the algorithm has a worst-case performance bound of 25/16.

3.2. Non-batch sequence-dependent setup times

The $1/ST_{sd}/\Sigma T_j$ problem was addressed in a number of papers. Tan and Narasimhan

(1997) proposed a simulated annealing algorithm and showed by computational analysis that it performs better than earlier algorithms. Different versions of genetic algorithms were proposed by Armentano and Mazzini (2000), Tan et al. (2000), and França et al. (2001). Gagne et al. (2002) proposed an Ant Colony Optimization (ACO) algorithm for the same problem and showed that it performs competitively with the best results of Tan et al. (2000) in terms of solution quality while it takes less computational time. Mendes et al. (2002a) presented a multi start procedure, which is a simple algorithm that creates an initial random solution and then applies a local search procedure repeatedly to the obtained initial solution. Gupta and Smith (2006) proposed two heuristics, a greedy randomized adaptive search procedure (GRASP) and a problem space-based local search heuristic. Gupta and Smith (2006) showed that the problem space-based local search heuristic performs equally well when compared to ACO of Gagne et al. (2002) while taking much less computational time. Gupta and Smith (2006) also showed that GRASP gives much better solutions than ACO while it takes much more computation time than ACO. However, the genetic versions of Armentano and Mazzini (2000), Tan et al. (2000), and França et al. (2001) remain to be compared with one another. Chang et al. (2004a) proposed a mathematical programming model with logical constraints for the problem of $1/ST_{sd},\ r_j/\sum w_jT_j$. They also proposed a heuristic algorithm and conducted computational experiments, which revealed that the heuristic can efficiently solve the problem.

For the common due date case, Rabadi et al. (2004) proposed a branch-and-bound algorithm for the $1/ST_{sd}/\sum E_j+\sum T_j$ problem, where they showed that problems with up to 25 jobs can be solved by the algorithm in a reasonable time. Wang and Wang (1997) considered the single machine earliness-tardiness scheduling problem, where they proposed a hybrid genetic algorithm with the objective of minimizing a penalty function that includes a penalty for completing a job early or tardy, and a penalty for the total setup time between the jobs. Miller et al. (1999) proposed a hybrid genetic algorithm for the problem with the objective of minimizing the sum of setup costs, in addition to inventory and backlog costs. They showed that the performance of the hybrid algorithm is much better than that of the pure genetic algorithm.

Eren and Guner (2006) considered the $1/ST_{sd}/\lambda\sum C_j+(1-\lambda)\sum T_j$ problem, where the objective is to minimize a weighted sum of total completion time and total tardiness. They developed an integer programming model for the problem. Moreover, they presented a simple heuristic and used this heuristic as an initial solution for their proposed Tabu search algorithm.

Asano and Ohta (1999) proposed a branch-and-bound algorithm for the $1/ST_{sd},\ r_j/T_{max}$ problem, where the machine may not be available for certain periods such as due to maintenance. They also developed a post-processing algorithm that manipulates the starting time of the shutdown period so as to reduce the obtained T_{max}. After obtaining an initial solution, Shin et al. (2002) presented a tabu search algorithm for the $1/ST_{sd},\ r_j/L_{max}$ problem. They showed that their algorithm obtains a much better solution than existing heuristics in less computational time.

Lee and Asllani (2004) presented a mixed integer programming and a genetic algorithm for the $1/ST_{sd}$ problem with the minimization of $\sum U_j$ as the primary objective, and the minimization of C_{max} as the secondary objective. They concluded that the integer programming becomes very complex and unmanageable when the number of jobs is more than ten. They also stated that computational analysis showed that the proposed genetic algorithm performs better when the ratio of setup times to processing times is relatively large.

Kolahan and Liang (1998) presented a tabu search approach to a just-in-time scheduling problem with sequence dependent setup times, in which there are linear costs for compression or extension of job processing times. The objective is a linear combination of the total weighted earliness and tardiness, and the total weighted compression and extension costs.

Koulamas and Kyparisis (2006) considered the single machine scheduling problem with setup times that are proportionate to the length of the already scheduled jobs, which they call past-sequence dependent setup times. They showed that the single machine problem with the objective functions of makespan, total completion time, total absolute differences in completion times, and a linear combination of the last two objective functions can be solved in *O(n long n)* time by a sorting procedure.

3.3. Batch sequence-independent setup times

Chen et al. (1997) considered the $1/ST_{si,b}$ problem with the objective of obtaining the optimal common due date and the optimal sequence of jobs to minimize the common due date cost (which is linearly related to the length of common due date) and the sum of the costs of tardy jobs. They addressed problems with and without the group technology assumption. For both cases, they presented properties of the optimal solutions, and they proposed algorithms to solve each problem in polynomial time.

Several authors addressed the $1/ST_{si,b}/L_{max}$ problem. Pan and Su (1997) developed several dominance properties and lower bounds, and utilized the properties and lower bounds in a branch-and-bound algorithm to solve the problem. Baker and Magazine (2000) pointed out that the problem size that can be solved depends on several factors such as the number of batches, the number of jobs in each batch, due date range, and setup factor. Pan et al. (2001) formulated the problem as an integer program and proposed a heuristic to solve the problem. Baker (1999) considered the case where the setup times are the same for different job families. He proposed and compared several heuristic procedures for the problem. The branch-and-bound algorithms of Pan and Su (1997), and Baker and Magazine (2000) remain to be compared, as well as the heuristics of Baker (1999) and Pan et al. (2001). Shufeng and Yiren (2002) modeled a practical steel pipe plant as the $1/ST_{si,b}/L_{max}$ problem with the group technology assumption, where the jobs are *a priori* partitioned into classes and the classes are grouped into families. A major setup occurs when the jobs are switched from one family to another, while a minor setup occurs when the jobs are switched from one class to another within the same family. They provided an integer programming formulation for the problem and proposed a tabu search-based heuristic.

The $1/ST_{si,b}/\sum U_j$ problem was shown to be NP-hard in the strong sense by Cheng et al. (2001a) even for the case where all the setup times and processing times are one unit. Liu and Yu (1999) proved that this problem is strongly NP-hard under the group technology assumption with unit processing times and zero setup times. The NP-hardness of the problem with the group technology assumption was also established by Liaee and Emmons (1997).

Pan and Wu (1998) considered the $1/ST_{si,b}/\sum f_j$ problem under the group technology assumption and the assumption that all the jobs are ready for processing at time zero. They proposed an algorithm to solve the problem, subject to the constraint that no jobs are tardy. The complexity of the algorithm is shown to have a polynomial running time in the number of groups and jobs. The $1/ST_{si,b}/\sum w_j f_j$ problem was addressed by Dunstall et al. (2000), where it is assumed that all the jobs are available from time zero. They developed lower bounds and incorporated these lower bounds into a branch-and-bound algorithm. Their algorithm is quite efficient since problems up to 70 jobs can be solved optimally within a reasonable time.

Yang and Chand (2006) addressed the $1/ST_{si,b}/\sum C_j$ problem, where the setup time of a batch is characterized by a learning factor and changes based on its position in a schedule. They developed two lower bounds, and implemented these lower bounds in a branch-and-bound algorithm. They concluded that the influence of learning on group scheduling increases with the speed at which a family accumulates experience.

Azizoglu and Webster (1997) proposed a branch-and-bound algorithm to solve the $1/ST_{si,b}/\sum w_j E_j+\sum w_j T_j$ problem with a common due date. They solved problems with up to 20 jobs and pointed out that problem size and weight combinations play a dominant role in the difficulty of obtaining optimal solutions. For large-sized problems, they presented a beam search procedure, which has a parameter by which a trade-off between the error and computation time can be made. Webster et al. (1998) proposed a genetic algorithm for the same problem. The results of their computational experiments showed that the algorithm converges close to optimal solutions quickly. Suriyaarachchi and Wirth (2004) provided several necessary conditions for a solution to be optimal. They also proposed a greedy heuristic and a genetic algorithm for the problem.

The $1/ST_{si,b}/D_{max}$ problem was addressed by Woeginger (1998), where each job has a delivery time. The best previously known polynomial time approximation algorithm for this problem has a worst-case guarantee of 3/2. Woeginger (1998) demonstrated the existence of a polynomial time approximation scheme for the problem.

Baptiste and Le Pape (2005) studied the $1/ST_{si,b}$, $SC_{si,b}$ problem with a sum of regular objective functions. The jobs may have different release dates and deadlines. They developed lower bounds and dominance properties, and proposed a branch-and-bound algorithm, which was evaluated experimentally.

The problem of $1/ST_{si,b}/\sum T_j$ was considered by Schaller (2006), where he proposed a branch-and-bound algorithm for the problem with and without the group technology assumption. He also proposed a heuristic to solve larger-sized problems. His computational experiments revealed that total tardiness can be significantly reduced by removing the group technology assumption. He solved problems with up to 10 families and 20 jobs in each family. Schaller and Gupta (2006) studied the same problem with the objective of $1/ST_{si,b}/\sum E_j + \sum T_j$ by following the same procedure.

All the job characteristics are assumed to be deterministic in all of the above literature. Van Oyen et al. (1999) addressed the problems of $1/ST_{si,b}/L_{max}$, $1/ST_{si,b}/\sum w_j f_j$, and $1/ST_{si,b}/\sum w_j T_j$, where the processing times and due dates are random variables, and the criterion is to minimize the expected value of the objective function. They considered the problem with and without the group technology assumption and derived conditions under which simple sequencing rules are optimal for each problem.

Assuming that all the jobs are ready at time zero, Liao and Liao (2002) considered the $1/ST_{si,b}/\sum f_j$ problem, where there are families of jobs and each family is partitioned into classes. A major setup time is required when processing is switched from one family to another, while a minor setup time is necessary when it is switched from one class to another. They proposed a tabu search algorithm for the problem and showed by computational analysis that their proposed algorithm performs better than the existing dynamic programming based heuristic.

Yuan et al. (2005a) showed that the $1/ST_{si,b}, r_j/C_{max}$ problem is strongly NP-hard even if the processing times of the jobs are unit and the setup times of the families are identical. They provided two dynamic programming algorithms, a heuristic with a performance ratio of 2, and a polynomial-time approximation scheme for the problem.

Agnetis et al. (2004) addressed a problem in which the jobs of the same family are processed in batches of the same size, each batch is preceded by a constant setup time and every job within each batch needs a sequence of specific tools, in which the tools can be repeated. A (super)sequence of all the required tools is loaded before the batch is processed. Each job uses a subsequence of the tools in this supersequence. The tools are used in parallel. The number of setups inside a batch is equal to the length of the corresponding tool supersequence minus one. The problem is to partition the jobs into the batches and, for each batch, determine a supersequence of the required tools such that the total number of setups is minimized. The authors suggested a heuristic algorithm for this problem.

Wagelmans and Gerodimos (2000) proposed an $O(nlogn)$ dynamic programming algorithm for a single family problem to minimize L_{max} under the batch availability model. Ng et al. (2002a) demonstrated that this problem with precedence constraints reduces in $O(n^2)$ time to the one without precedence constraints.

A single family problem with equal job processing times and arbitrary job release dates was studied by Baptiste (2000) under the batch availability model and non-anticipatory setups for various regular objectives. Dynamic programming algorithms of $O(n^{14})$ running time were derived for minimizing Σw_jU_j, Σw_jC_j and ΣT_j, and T_{max} was minimized in $O(n^{14}logn)$ time. Ng et al. (2003b) improved this result for the ΣC_j objective by presenting an $O(n^5)$ time algorithm even if there are precedence constraints. Yuan et al. (2004) showed that the above problem to minimize L_{max} with precedence constraints reduces in $O(n^2)$ time to the one without precedence constraints.

For a single family problem with equal job processing times and common setup time to minimize ΣC_i under the batch availability model, Mosheiov et al. (2005) suggested an $O(n)$ rounding procedure to calculate integer batch sizes from a straightforward solution of the relaxed non-integer batching problem. Mosheiov and Oron (2006a) extended these results to the case where batch sizes are bounded from below or above.

Baptiste and Jouglet (2001) suggested a pseudo-polynomial dynamic programming algorithm for a single family problem to minimize the total tardiness. Cheng et al. (2001b) developed polynomial time algorithms for two single family problems with job processing times and setup times dependent on two different uniform resources. The batch availability model was considered. In one problem, the objective is to minimize the total weighted resource consumption, subject to meeting job deadlines, and in the other problem, the objective is to minimize L_{max}, subject to an upper bound on the total weighted resource consumption. The algorithms are based on solving an integer linear program with two variables.

Cheng and Kovalyov (2001) studied a single family problem under the batch availability model for various objective functions. Properties of optimal schedules were established and polynomial-time dynamic programming algorithms were derived for the cases where there are a constant number of distinct processing times or a constant number of distinct due dates. The same model was considered by Hochbaum and Landy (1997), where the job processing times are all equal and the job weights take at most two distinct values, and the objective is to minimize Σw_jC_j. An $O(\sqrt{n}\log n)$ time algorithm was suggested.

Cheng et al. (2003a) proved the strong NP-hardness of the problem $1/ST_{si,b}/L_{max}$. Schultz et al. (2004) proposed a neighborhood search heuristic for this problem.

Dang and Kang (2004) presented a 2-approximation algorithm for a single family problem, in which the setup time for a batch is given by the maximum job setup time in this batch and the processing time of a batch is given by the maximum job processing time in this batch. The objective is to minimize the total weighted completion time. Computational complexity of this problem is unknown.

Yuan et al. (2005b) addressed the $1/ST_{si,b}=s/\sum w_jC_j$ (where $ST_{si,b}=s$ means constant setup times) problem with the restriction that each batch contains the same number of jobs (called the batch size). They proved that this problem is strongly NP-hard even if the batch size is 3 and the weight of each job is equal to its processing time. $O(nlogn)$ time algorithms were given for two special cases of the problem.

Gerodimos et al. (2000) studied a problem in which each job consists of two components: standard and specific. Standard components are processed in batches under the batch availability model. Each batch is preceded by a constant setup time. A job is completed when both its components have been completed. For any regular objective function, Gerodimos et al. proved that there exists an optimal schedule in which the specific components of the jobs in the same batch (of standard components) immediately follow this batch. Therefore, the problem reduces to finding a sequence of specific components and its partition into subsequences corresponding to the batches of standard components. The earliest due date (EDD) sequence was proved to be optimal for L_{max} minimization and for the early jobs in case of $\sum U_j$ minimization. The latter problem was proved to be NP-hard. Both problems were solved by dynamic programming algorithms in $O(n^2)$ and $O(n^2 d_{max})$ time, respectively, where d_{max} is the maximum due date. Wagelmans and Gerodimos (2000) improved the algorithm for L_{max} minimization to have $O(nlogn)$ time complexity. Yang (2004b) generalized the model of Gerodimos et al. (2000) by assuming that the common components belong to several families and a sequence independent setup time precedes a batch of such components. For $\sum C_j$ minimization, Yang gave some properties of an optimal solution and suggested a branch-and-bound algorithm, which was shown to be able to solve problems with up to 5 families and 40 jobs in each family.

Gerodimos et al. (2001) and Lin (2002) studied a problem differing from the problem of Gerodimos et al. (2000) in that the job availability model is applied for batch processing of the standard components. For L_{max} and $\sum U_j$ minimization problems, Gerodimos et al. (2001) obtained the same results as those under the batch availability model with respect to computational complexity. Furthermore, an $O(nlogn)$ time algorithm was developed for the problem of minimizing $\sum C_j$ in the case of agreeable processing times between the standard and specific operations (they can be similarly ordered). These results outperformed those of Lin (2002).

Gerodimos et al. (1999) studied a problem in which each job consists of up to F operations belonging to different families. A job is completed when all its operations have been processed. A sequence independent setup time occurs between the operations of different families. Like under the job availability model, operations are completed individually. The problem of minimizing L_{max} was shown equivalent to the problem $1/ST_{si,b}/L_{max}$. Therefore, it is NP-hard in the strong sense and solvable in polynomial time for any fixed F. Minimization of $\sum U_j$ was proved to be NP-hard in the strong sense and pseudopolynomially solvable for any fixed F. Cheng et al. (2003b) proved that this problem of minimizing $\sum U_j$ remains strongly NP-hard even if the due-dates are the same and all the jobs have the same processing time. Ng et al. (2002b) proved that the problem of minimizing $\sum C_j$ is strongly NP-hard even if the setup times are the same and each operation processing time is 0 or 1. It is polynomially solvable if the operation processing times are all agreeable and F is fixed (Gerodimos et al. 1999).

Yang and Liao (1998) suggested a branch-and-bound algorithm for a problem in which each job is attributed to a family and an order. There is a sequence independent setup time between the jobs of different families. An order is completed upon the completion of its latest job. The objective is to minimize the total order completion time. Problems up to 24 jobs were solved by the branch-and-bound algorithm.

Multiple family problems were studied by Janiak et al. (2005) and Ng et al. (2005) under the group technology assumption with resource dependent setup and processing times. It is assumed that the same amount of one resource is assigned to all the setups and the same amount of another resource is assigned to all the jobs. The resources can all be continuously divisible or all discrete. Janiak et al. (2005) presented polynomial-time algorithms based on geometric techniques to minimize the total weighted resource consumption, provided that the job deadlines are met. Ng et al. (2005) derived polynomial-time algorithms to minimize $\sum w_jC_j$, subject to an upper bound on the total weighted resource consumption. A key element of the algorithms is a reduction to solving a linear programming problem with two variables.

A similar solution approach was used by Ng et al. (2004) to handle a single family problem under the batch availability model. Polynomial-time algorithms were derived for minimizing $\sum C_j$, subject to an upper bound on the total weighted resource consumption and an inverse problem (minimizing the total weighted resource consumption, subject to an upper bound on $\sum C_j$).

Ng et al. (2003a) extended the model of Ng et al. (2004) by allowing job and batch dependent resource consumptions. They mentioned that the computational complexity of the problem of minimizing $\sum C_j$, subject to an upper bound on the total weighted resource consumption, is unknown and developed polynomial-time algorithm for this problem and an inverse problem in the case where lower and upper bounds on the job processing times are agreeable (can be similarly ordered).

Soric (2000a, 2000b) and Vieira et al. (2000) studied a problem with dynamically arriving jobs belonging to a fixed number of families. There is a constant setup time s between the jobs of different families. Soric considered the objective of minimizing the average work backlog. In the case of an infinite number of jobs, Soric (2000a) suggested an on-line heuristic called Clear-the-Largest-Work-after-Setup, which chooses for production at a decision time point t a family with the largest work backlog at time $t+s$. If a family is chosen for production, all the jobs of this family having arrived so far are produced. In the case of a finite number of jobs, Soric (2000b) developed a mixed integer linear programming formulation and a cutting plane branch-and-bound algorithm.

Vieira et al. (2000) considered a stochastic environment with machine breakdowns. They suggested a rescheduling algorithm and analytically compared its performance with respect to the average flowtime, machine utilization, setup frequency and rescheduling frequency under periodic and event-driven rescheduling strategies. In their study, periodic rescheduling occurs every h time units, while event-driven rescheduling occurs when a new job arrives. The suggested algorithm groups unprocessed jobs of the same family into the same batch, dispatches jobs of the same batch according to the First-In-First-Out (FIFO) rule and dispatches batches according to the FIFO rule applied to the first jobs of the batches.

Kuik and Tielemans (1997) studied the effect of batch sizes on setup utilization (total setup time divided by total setup and processing time) in a queuing delay batching model. They established an upper bound of $3-2\sqrt{2}\approx 0.175$ on the optimal setup utilization. This result implies that batch sizes should be corrected if setup utilization is higher than 0.175 for the considered model.

Tovey (2004) considered the problem with multiple families and arbitrary precedence relations with the objective of minimizing the number of setups. He proved that the objective cannot be approximated in polynomial time with a constant worst-case performance ratio unless P=NP.

Wang and Zou (2002) studied a steel pipe plant scheduling problem, where the jobs are partitioned into classes and the classes are grouped into families. A major setup time is required when jobs are switched from one class to another while a minor setup time is necessary when jobs are switched from one family to another within the same class. Under the group technology assumption with regard to families, Wang and Zou (2002) proposed a mixed integer programming, and presented a tabu search heuristic to solve the problem with respect to the maximum lateness criterion.

3.4. Batch sequence-dependent setup times

The $1/ST_{sd,b}/C_{max}$ problem was addressed by Van Der Veen et al. (1998), where they modeled the problem as an asymmetric traveling salesman problem with a specific distance matrix. They proposed a polynomial-time algorithm to solve the problem.

Sun et al. (1999) studied the $1/ST_{sd,b}, r_j/\sum w_j T_j^2$ problem. They developed a Lagrangian relaxation based approach for the problem, where the setup times are treated as capacity constraints. They compared the performance of this approach with several other procedures, including tabu search and simulated annealing.

Karabati and Akkan (2006) presented a branch-and-bound algorithm for the $1/ST_{sd,b}/\sum C_j$ problem, where they developed a lower bound that is based on a network formulation of the problem. Computational analysis showed that problems with up to 60 jobs and 12 families can be solved optimally using the algorithm.

Sourd (2005) addressed the general problem of $1/ST_{sd,b}, SC_{sd,b}$ with the objective of minimizing the earliness-tardiness and setup costs. He proposed a mixed integer formulation from which lower bounds were derived and used in a branch-and-bound algorithm, which can solve problems up to 20 jobs. He also proposed a heuristic to solve large-sized problems.

Gupta and Sivakumar (2005) considered the $1/ST_{sd,b}$ problem with the multiple objectives of minimizing the average tardiness and cycle time, and maximizing the machine utilization. They proposed an approach that generates a Pareto optimal solution for the problem.

4. Parallel Machines

There are m machines in parallel, where machines may be identical (P), or have different speeds or uniform (Q), or completely unrelated (R). Each job can be performed on any of the machines. A summary of the setup time (cost) literature in this environment is given in Tables 3 and 4, where the uniform (unrelated) machines are indicated by the letter Q(R) in the third column in the "Comments" area. If there is no letter of Q or R in this area, which is the vast majority of the cases, it means that the machines are identical.

4.1. Non-batch sequence-independent setup times

There is no need to consider non-batch sequence-independent setup times for the general parallel-machine problem, since the setup times can be included in the processing times. However, for certain parallel-machine problems, the setup times should be considered as separate from the processing times. It is assumed that a job can be processed by at most one machine at a time for the general parallel-machine problem. Xing and Zhang (2000) considered the case where a job can be processed on two different machines at the same time. For this problem, Xing and Zhang (2000) presented a heuristic with a worst-case performance ratio of $7/4-1/m$ ($m \geq 2$) when the objective function is C_{max} and the jobs have sequence-independent setup times. Assuming that a setup is required each time a job is preempted, Schuurman and Woeginger (1999) proposed an approximation algorithm for the $P/ST_{si}, pmtn/C_{max}$ problem, where the worst case ratio of the algorithm can be made arbitrarily close to 4/3. They also demonstrated the existence of a polynomial-time algorithm for the case of equal setup times.

The following parallel-machine scheduling problem has recently been addressed in the literature. There is a set of n jobs to be processed on a set of m parallel machines. The loading of a job on a machine, the time of which is called the setup time, is performed by a single server. This setup time cannot be performed while a machine is processing a job. On the other hand, the machine can process a job without the server being present after the job is loaded on the machine. Simultaneous requests of the server by the machines will result in machine idle time. This problem is denoted as $P,S/ST_{si}/\gamma$. When the setup time is constant for each job, it is denoted as $ST_{si}=s$.

Kravchenko and Werner (1997) studied the $P,S/ST_{si}=s/C_{max}$ problem and showed that it is strongly NP-hard. They analyzed some list scheduling heuristics for the problem and presented some polynomially solvable cases. Kravchenko and Werner (2001) considered the same problem but with unit setup times and the total completion time criterion, i.e., $P,S/ST_{si}=1/\sum C_j$. They presented a heuristic algorithm and proved that the heuristic has an absolute error bounded by the product of the number of short jobs (with processing times less than $m-1$) and $m-2$. Wang and Cheng (2001) addressed the $P,S/ST_{si}/\sum w_jC_j$ problem and presented a $(5-1/m)$ approximation algorithm, which is based on a linear relaxation. They also showed that the SPT (Shortest Processing Time) schedule is a 3/2 approximation for the $P,S/ST_{si}=s/\sum C_j$ problem.

The problem was also addressed for the case of two parallel machines. Koulamas (1996) showed that the $P2,S/ST_{si}$ problem with the objective of minimizing the machine idle time resulting from unavailability of the server is NP-hard in the strong sense. He proposed an efficient beam search heuristic for the problem. Abdekhodaee and Wirth (2002) addressed the $P2,S/ST_{si}/C_{max}$ problem under a specific assumption of *alternating job processing,* the definition of which was not precisely given. They proved the strong NP-hardness of the problem, suggested an integer programming formulation, and presented polynomial algorithms for several more restricted cases. Abdekhodaee et al. (2004) considered the same problem, but for the special cases of equal processing and setup times. They proved the NP-hardness of two special cases of the same problem, and proposed heuristics for each case. Abdekhodaee et al. (2006) also considered the same problem for the general case. They proposed greedy heuristics and a genetic algorithm for the general case. They also proposed the use of the well-known Gilmore-Gomory algorithm to solve the general case.

Hall et al. (2000) proved the strong NP-hardness of the problems $P2,S/ST_{si}$ with C_{max} and $\sum C_j$ objectives for the case where the setup times are all equal. If all the job processing times are one unit, then the problem $P2,S/ST_{si}$ with the objectives $\sum T_j$ and $\sum w_jU_j$ is NP-hard, with the objective $\sum w_jT_j$ is strongly NP-hard, and the problem $P,S/ST_{si}$ with the objectives C_{max}, L_{max}, $\sum C_j$, $\sum w_jC_j$ and $\sum U_j$ is polynomially solvable. The questions about the NP-hardness of the problem $P,S/ST_{si}/\sum C_j$ and the strong NP-hardness of the problem $P2,S/ST_{si}/\sum w_jC_j$ were left open. The first question was answered by Brucker et al. (2002), who proved the strong NP-hardness of the problem $P,S/ST_{si}/\sum C_j$. Brucker et al. derived numerous complexity results for the server scheduling problems in the parallel-machine environment. They made and proved an observation that several classical single- and parallel-machine scheduling problems polynomially reduce to their server counterparts. The NP-hardness of a number of server scheduling problems readily follows from this reduction. They developed an $O(n^7)$ algorithm for the problem $P3,S/ST_{si}/\sum C_j$ with unit setup times and a number of polynomial algorithms for the special cases with equal setup times and equal processing times. The complexity of the problem $Pm,S/ST_{si}/\sum C_j$ with given $m \geq 4$ has remained unsolved. Guirchoun et al. (2005) presented more complexity results for the problem.

Another comprehensive paper was by Glass et al. (2000), who addressed the same problem with the C_{max} objective function but with dedicated machines, where each machine processes its own set of pre-assigned jobs. In other words, the set of n jobs is in advance split into m subsets, where all the jobs in a subset are performed by the same machine. They proved that the problem with two dedicated machines is NP-hard in the strong sense even if all the setup times are equal or if all the processing times are equal. They showed that a simple greedy algorithm creates a schedule that is at most twice the optimal value for the case of m machines. They also presented a heuristic with a worst-case ratio of 3/2 for the case of two machines.

In all of the above mentioned research, the setup times were assumed to be performed by a single server. Kravchenko and Werner (1998) considered the problem with *m-1* servers where there are *m* machines. They presented a pseudopolynomial-time algorithm for the problem when the objective is to minimize C_{max}.

4.2. Non-batch sequence-dependent setup times

Heady and Zhu (1998) addressed the P/ST_{sd} problem, where some machines may not be able to process some jobs. They proposed a heuristic to minimize the sum of earliness and tardiness costs for the problem. For small-sized problems, they also compared the performance of the proposed heuristic with the optimal solution obtained from using integer programming formulation. Vignier et al. (1999) considered the $P/ST_{sd},r_j$ problem, where there are two types of machines, both processing and setup times depend on the machines, and each job has a release date and a due date. The objective is to find a feasible schedule first and then to minimize the cost due to assignment and setup times. They proposed a hybrid method that consists of an iterative heuristic, a genetic algorithm, and a branch-and-bound algorithm.

Radhakrishnan and Ventura (2000) addressed the $P/ST_{sd}/\sum E_j+\sum T_j$ problem, presented a mathematical programming formulation that can be used for limited-sized problems, and proposed a simulated annealing algorithm for large-sized problems. Feng and Lau(2005) addressed the more general $P/ST_{sd}/\sum w_jE_j+\sum w_jT_j$ problem and proposed a meta-heuristic called Squeaky Wheel Optimization. Feng and Lau (2005) showed that their heuristic outperforms that of Radhakrishnan and Ventura. Hiraishi et al. (2002) considered the P/ST_{sd} problem with the objective of maximizing the weighted number of jobs that are completed at their due dates. They showed that some special cases of the problem are polynomially solvable while the problem is NP-hard in general.

Mendes et al. (2002b) and Gendreau et al. (2001) addressed the $P/ST_{sd}/C_{max}$ problem. Medes et al. (2002b) proposed two heuristics, namely one tabu search based and the other a memetic approach that is a combination of a population based method with local search procedures. Gendreau et al. (2001) proposed lower bounds and presented a divide and merge heuristic. They compared their heuristic with earlier heuristics of tabu search and showed that their heuristic is much faster while producing similar quality results. Tahar et al. (2006) addressed the same problem of $P/ST_{sd}/C_{max}$ with job splitting. Job splitting is different from preemption in that jobs can be split and processed simultaneously on different machines. They proposed a heuristic based on linear programming modeling. The performance of their proposed method was tested on problems of different sizes by comparing the solutions of the method with a lower bound.

Hurink and Knust (2001) addressed the $P/ST_{sd},prec/C_{max}$ problem, where they considered the problem as a combination of two parts, namely partitioning and sequencing. They established that the problem is strongly NP-hard, where the starting times respect a given order for the case of no precedence relations. Fixing the sequencing problem first, they showed that it is unlikely that an efficient list scheduling algorithm exists that leads to a dominant set of schedules. As a result, they concluded that the problem cannot be solved by considering only the decisions for one of its two parts as the solution space and solving the remaining sub-problem afterwards. Kurz and Askin (2001) presented an integer programming formulation for the problem of $P/ST_{sd},r_j/C_{max}$. They also developed several heuristics including genetic algorithms and multi-fit based approaches and empirically evaluated them. They used solution of the traveling salesman problem (TSP) as part of their heuristics. That is, once the jobs have been assigned to the machines, a TSP is formulated and solved to find an optimal job sequence on each machine. In the TSP, the (asymmetric) distances correspond to the (sequence dependent) setup times. Kim and Shin (2003) proposed a restricted tabu search algorithm for the $P/ST_{sd},r_j/L_{max}$ problem for both cases of identical and non-identical machines. The restricted search algorithm reduces the search effort significantly without eliminating promising solutions.

Weng et al. (2001) addressed the $R/ST_{sd}/\Sigma w_jC_j$ problem. They presented seven simple heuristics for the problem and showed by computational experiments that one of them outperforms the others. The best heuristic assigns one job at a time based on the smallest ratio of a job's processing time plus setup time to its weight. Fowler et al. (2003) proposed a hybrid genetic algorithm for the $P/ST_{sd}, r_j/\Sigma w_jC_j$, $P/ST_{sd}, r_j/\Sigma w_jT_j$, and $P/ST_{sd}, r_j/C_{max}$ problems. In the hybrid genetic algorithm, a genetic algorithm is used to assign jobs to machines, and dispatching rules are used to schedule the individual machines. Computational results indicated that the proposed hybrid approach performs better than earlier algorithms with respect to the considered performance measures. The $P/ST_{sd}, r_j/\Sigma C_j$ problem was addressed by Nessah et al. (2005). They presented a necessary and sufficient condition for a local optimal solution and proposed a heuristic that is based on the condition. They also developed a lower bound. The quality of their heuristic was tested on randomly generated problems by comparing the heuristic solution with a developed lower bound. Clearly, the genetic algorithm of Fowler et al. (2003) and the heuristic proposed by Nessah et al. (2005) remain to be compared.

Tamimi and Rajan (1997) proposed a genetic algorithm for the $Q/ST_{sd}/\Sigma w_jT_j$ problem. In their genetic algorithm, they dynamically modified the mutation rate, crossover rate, and insertion rate. Park et al. (2000) proposed the use of a neural network to obtain values for the parameters in calculating a priority rule for the $P/ST_{sd}/\Sigma w_jT_j$ problem. Their computational results indicated that their proposed approach outperforms that of an earlier approach. Kim et al. (2003b) presented a heuristic for the same problem, which consists of four phases, where the third phase is a tabu search. A comparison of the genetic algorithm presented by Tamimi and Rajan (1997) and the hybrid heuristic proposed by Kim et al. (2003b) remains to be performed. Bilge et al. (2004) presented a tabu search algorithm for the $P/ST_{sd}, r_j/\Sigma T_j$ problem. They investigated several key components of tabu search and identified the best values for these components. They compared their heuristic with the genetic algorithm of Sivrikaya-Serifoglu and Ulusoy (1999) for the case of zero weight for earliness, and showed that their heuristic outperforms that of Sivrikaya-Serifoglu and Ulusoy (1999).

Sivrikaya-Serifoglu and Ulusoy (1999) addressed the problem of $Q/ST_{sd}, r_j/w_E\sum E_j+w_T\sum T_j$, where there are two types of machines with different speeds. Here $w_E\sum E_j+w_T\sum T_j$ means that the weights for earliness and tardiness penalties are common to all the jobs. Sivrikaya-Serifoglu and Ulusoy (1999) presented two types of genetic algorithms, namely one with a crossover operator and one without crossover operator. They showed that the genetic algorithm with a crossover operator performs better for difficult and large-sized problems. Balakrishnan et al. (1999) considered the general case of uniform machines with the objective function of minimizing $\sum w_jE_j+\sum w_jT_j$. They presented a mixed integer programming formulation for the problem. Zhu and Heady (2000) addressed the $R/ST_{sd}/\sum w_jE_j+\sum w_jT_j$ problem. They developed a mixed integer programming formulation for the problem, which can provide an optimal solution in reasonable time for nine jobs and three machines.

The P/SC_{sd} problem with the objective of minimizing the total setup costs was considered by Anglani et al. (2005). They considered the case where the job processing times are uncertain, and proposed a fuzzy mathematical programming approach to solve the problem. They also showed that the problem can be converted into a mixed integer linear programming model. Moreover, they proposed an approximation model that can be used to handle larger problems and showed that the average deviation of the approximation model solution over the optimal solution is less than 1.5%.

4.3. Batch sequence-independent setup times

Liu et al. (1999) proved the ordinary NP-hardness and presented a pseudopolynomial-time algorithm for the multiple family problem $P2/ST_{si,b},p_j=p/\sum C_j$ with a common setup time. Liaee and Emmons (1997) proved the ordinary NP-hardness of the same problem under the group technology assumption unless all the families contain the same number of jobs. Blazewicz and Kovalyov (2002) proved the strong NP-hardness of the problem $P/ST_{si,b}/\sum C_j$ under the group technology assumption, and presented a polynomial-time dynamic programming algorithm for the special case with a given number of the machines.

Leung et al. (2006) considered the problem $Pm/ST_{si,b}=s/\sum C_j$, where the processing time of each job is a step function of its waiting time, i.e., the time between the start of the processing of the batch to which the job belongs and the start of the processing of the job. For each job i, if its waiting tmie is less than a given threshold D, then it requires a basic processing time $p_i=a_i$; otherwise, it requires an extended processing time $p_i=a_i+b_i$. They proved that this problem is NP-hard in the strong sense, even if there is only one machine and $b_i=b$ for all $i=1,\dots,n$; and is polynomially solvable, if $b_i=b$ for all $i=1,\dots,n$. An approximation algorithm with performance guarantee 2 was given for the case $b_i \leq D$, $i=1,\ldots,n$.

Yi and Wang (2001a) proposed a tabu search algorithm, while Yi and Wang (2001b) presented a lower bound for the $P/ST_{si,b}/\Sigma f_j$ problem with the assumption that the jobs are ready at time zero. Yi et al. (2004) proposed a fuzzy logic embedded genetic algorithm for the same problem. Webster and Azizoglu (2001) and Azizoglu and Webster (2003) addressed the same problem with a weighted objective function, i.e., $P/ST_{si,b}/\Sigma w_j f_j$, or equivalently $P/ST_{si,b}/\Sigma w_j C_j$. Two dynamic programming algorithms (a backward and a forward) were proposed by Webster and Azizoglu (2001), where they also identified the characteristics of the problems for which each algorithm is suitable. When the number of machines and families are fixed, the backward dynamic algorithm is polynomial in the sum of the weights while the forward dynamic algorithm is polynomial in the sum of processing and setup times. Azizoglu and Webster (2003) presented several branch-and-bound algorithms for the problem and computationally evaluated the performance of each algorithm. They concluded that the algorithms can quickly generate optimal solutions for problems with up to 15 to 25 jobs, depending on the number of machines. Chen and Powell (2003) proposed column generation based branch-and-bound algorithms for the same problem, where they obtained optimal solutions for problems up to 40 jobs, 4 machines and 6 families. Dunstall and Wirth (2005a) presented another branch-and-bound algorithm for the same problem, and they showed that their algorithm outperforms that of Azizoglu and Webster (2003). They solved problems with up to 25 jobs and 8 families using their branch-and-bound algorithm. Dunstall and Wirth (2005b) proposed several simple heuristics for the same problem. Clearly, the branch-and-bound algorithms of Chen and Powel (2003) and of Dunstall and Wirth (2005a) remain to be compared. Also, the heuristics of Yi et al. (2004) and of Dunstall and Wirth (2005b) remain to be compared for at least the same weight of all the jobs since Yi et al. (2004) considered the case of non-weighted jobs.

Chen and Powell (2003) proposed column generation based branch-and-bound algorithms for the $P/ST_{si,b}/\Sigma w_j U_j$ problem. They obtained optimal solutions for problems with up to 40 jobs, 6 families and 4 machines. Chen and Wu (2006) addressed the $R/ST_{si,b}/\Sigma T_j$ problem and proposed a heuristic based on threshold-accepting methods, tabu list, and improvement procedures. They showed by computational analysis that the heuristic significantly outperforms a simulated annealing heuristic. Yi and Wang (2003) considered the $P/ST_{si,b}/\Sigma w_j E_j + \Sigma w_j T_j$ problem, where the jobs have a common due date. They proposed a fuzzy logic embedded genetic algorithm (called soft computing) to solve the problem.

Gambosi and Nicosia (2000) proposed an on-line algorithm for the $P/ST_{si,b}/C_{max}$ problem and derived an upper bound on its competitive ratio. They also derived a lower bound on the competitive ratio for any on-line algorithm. Crauwels et al. (2006) proposed an integer programming formulation and several heuristics for the $P/ST_{si,b}$, r_j, d_j problem for a number of performance measures including minimization of the number of setups.

Cheng and Kovalyov (2000) studied a single family problem of scheduling jobs by their deadlines on unrelated parallel machines under the batch availability model. Each batch is preceded by a constant setup time. They suggested a dynamic programming algorithm and an approximation scheme with $O(n^{2m+1}/\varepsilon^m)$ running time. The scheme delivers a schedule satisfying $C_j \leq (1+\varepsilon)d_j$ for all the jobs if a feasible (with respect to the due dates) schedule exists. The case of uniform machines and identical jobs was proved to be strongly NP-hard and solvable in $O(m^2n^{2m+1})$ time. If the machines are identical and job processing times are all equal to the setup time, the problem can be solved in $O(nlogn)$ time.

The single family problems $P/ST_{si,b}/L_{max}$ and $P/ST_{si,b}/\Sigma U_j$ with common batch setup time were studied by Lin and Jeng (2004) under the batch availability model. Dynamic programming algorithms that are pseudopolynomial for a fixed number of machines were presented, as well as heuristics based on the smallest completion time first and smallest lateness first rules.

Wilson et al. (2004) studied the problem $P/r_j, ST_{si,b}/C_{max}$ with a common batch setup time motivated by planning of cut and sew operations in upholstered furniture manufacturing. They suggested a batch splitting and scheduling heuristic and integrated this heuristic into a genetic algorithm.

Similar to Gerodimos et al. (2000) for a single machine problem, Yang (2004a) studied a parallel-machine problem in which each job consists of two components: standard and specific to be processed in this order on the same machine. Standard components are processed in batches under the batch availability model. Each batch is preceded by a constant setup time. A job is completed when both of its components are completed. For ΣC_j minimization, Yang proposed two constructive heuristics.

4.4. Batch sequence-dependent setup times

Kim et al. (2002) addressed the $R/ST_{sd,b}/\Sigma T_j$ problem, where the jobs in the same family have the same due date. They proposed a simulated annealing algorithm that utilizes job rearranging techniques to generate neighborhood solutions. They indicated by computational analysis that the simulated annealing algorithm outperforms a neighborhood search method. Eom et al. (2002) proposed a three-phase heuristic for the $P/ST_{sd,b}/\Sigma w_jT_j$ problem. Tabu search is used in the final phase of the algorithm. A comparison of the simulated annealing algorithm of Kim et al. (2002) and the heuristic of Eom et al. (2002) remains to be performed for at least the case with equal weights and identical machines since Eom et al. (2002) considered the identical machines case. The $P/ST_{sd,b}, r_j/\Sigma T_j$ problem was addressed by Dupuy et al. (2005) for the case involving the so-called calendar constraints. They presented a simulated annealing heuristic by introducing several neighborhood mechanisms. By computational experiments they showed that their proposed neighborhood mechanisms produce better results in a shorter time compared with several greedy heuristics and a basic simulated annealing procedure.

A generalization of the problem $R/ST_{sd,b}/\Sigma w_jT_j$ was studied by Kim et al. (2003a). In this problem, machines are classified into groups of identical machines. Each job consists of the same number of operations that can be processed simultaneously on different machines. A job is completed when its last operation is finished. Operation processing time depends on the job and the machine group. Job weights are inversely proportional to job due dates. A sequence dependent setup time occurs between batches of operations of different jobs. The authors presented and computationally tested several constructive heuristics: earliest weighted due date and shortest weighted processing time sequencing rules, specific batching heuristic and simulated annealing, using some real problems from semiconductor manufacturing.

Yalaoui and Chu (2003) proposed a heuristic algorithm for a modification of the problem $P/ST_{sd,b}/C_{max}$, in which a job can be split into several parts allowable to be processed in parallel. A reduction to the traveling salesman problem was used in the heuristic.

Chen and Powell (2003) proposed column generation based branch-and-bound algorithms for the $P/ST_{sd,b}/\sum w_jC_j$ and $P/ST_{sd,b}/\sum w_jU_j$ problems. Computational analysis showed that the algorithms are capable of optimally solving problems of medium size, i.e., up to 40 jobs, 4 machines and 6 families.

In paper manufacturing, Akkiraju et al. (2001) observed a model generalizing the $R/ST_{sd,b}$ problem with multiple objectives such as $\sum w_jT_j$, $\sum w_jE_j$, and TST. They suggested a heuristic approach based on the so-called Asynchronous Team architecture. Initial solutions are first generated by different experts and computer programs. Then these solutions are perturbed and improved. Finally, a set of Pareto optimal solutions is presented to a decision maker.

Jeong et al. (2001) studied a generalization of the $R/ST_{sd,b}/\sum f_j$ problem observed from the Thin Film Transistor Liquid Crystal Display (TFT LCD) assembly process. The objective is a linear combination of the mean flowtime and deviation from product demand. Two specific constructive heuristic algorithms were developed.

5. Flow Shops

In an m-machine flow shop, there are m stages in series, where there exist one or more machines at each stage. Each job has to be processed in each of the m stages in the same order. That is, each job has to be processed first in stage 1, then in stage 2, and so on. Operation times for each job in different stages may be different. We classify flow shop problems as (i) flow shop (there is one machine at each stage), (ii) no-wait flow shop (a succeeding operation starts immediately after the preceding operation completes), (iii) flexible (hybrid) flow shop (more than one machine exist in at least one stage), and (iv) assembly flow shop (each job consists of m-1 specific operations, each of which has to be performed on a pre-determined machine of the first stage, and an assembly operation to be performed on the second-stage machine).

5.1. Non-batch sequence-independent setup times

5.1.1 Flow shop

Assuming that jobs are ready at time zero, Allahverdi (2000) addressed the $F2/ST_{si}/\sum f_j$ problem, where he obtained optimal (analytical) solutions for certain cases, and established two dominance relations for the general problem. Moreover, he proposed a branch-and-bound algorithm by which problems with up to 35 jobs can be solved optimally in reasonable time. He also proposed three heuristics and compared them with one another.

In order to enable end users to be connected to local or remote databases (Intranet/ Internet) through the Web, a robust and scalable model is required to provide an interface between the enterprise service and clients. A model that is rapidly spreading uses two separate servers, an application server and a database server. This model is commonly known as the three-tiered architecture. Al-Anzi and Allahverdi (2001) showed that the three-tiered client-server database internet connectivity problem is equivalent to the $F2/ST_{si}/\sum f_j$ problem. Therefore, the results of Allahverdi (2000) can be used for this problem. Moreover, Al-Anzi and Allahverdi (2001) proposed nine additional heuristics for the problem, and showed that their proposed heuristics outperform those of Allahverdi (2000). Allahverdi and Aldowaisan (2002) also considered the same problem but with the removal times separated from the processing times in addition to setup times, i.e., $F2/ST_{si},R_{si}/\sum f_j$. They obtained analytically optimal solutions for special cases when the setup, processing and removal times satisfy certain conditions. They also developed dominance relations, a lower bound, and a branch-and-bound algorithm for the general problem. The branch-and-bound algorithm yields optimal solutions for up to 35 jobs. Moreover, they proposed different heuristics for the problem.

Allahverdi and Al-Anzi (2006a) studied the $F3/ST_{si}/\sum C_j$ problem. They developed a lower bound, an upper bound, and a dominance relation. Moreover, they presented a branch-and-bound algorithm for the problem, where problems up to 18 jobs can easily be solved.

Allahverdi and Al-Anzi (2002) showed that the multimedia data objects scheduling problem for WWW applications can be modeled as $F2/ST_{si}/L_{max}$. They established dominance relations and proposed four heuristics that outperform the existing ones for the problem. Many dominance relations have been established in the literature on scheduling problems, which are mainly used in implicit enumeration techniques to further reduce the search space for an optimal solution. Al-Anzi and Allahverdi (2006) proposed a novel method for discovering dominance relations for any scheduling problem. After the description of the method, they applied it to the $F2/ST_{si}/L_{max}$ problem. They analyzed the performance of the dominance relations they obtained by the proposed method, as well as the dominance relations proposed earlier including those of Allahverdi and Al-Anzi (2002). Allahverdi et al. (2005) proposed a hybrid genetic algorithm for the same problem, and showed that their proposed algorithm outperforms those of Allahverdi and Al-Anzi (2002). Ng et al. (2006) presented a dominance relation and several heuristics for the $F3/ST_{si}/L_{max}$ problem.

Fondrevelle et al. (2005b) studied the permutation m-machine flow shops with exact time lags to minimize L_{max}, where the case of negative time lags corresponds to job overlapping, which can be used to model the sequence independent setup time problem. They studied polynomial special cases and provided a dominance relation. They also derived lower and upper bounds and presented a branch-and-bound algorithm.

Al-Anzi and Allahverdi (2005a) addressed the Fm/ST_{si} problem with the objective of minimizing the completion time variance. They presented a hybrid evolutionary heuristic and showed by computational analysis that their heuristic outperforms previous heuristics.

Su and Chou (2000) addressed the $F2/ST_{si}$ problem with the objective of minimizing a weighted sum of C_{max} and $\sum f_j$ in a dynamic environment, where jobs keep arriving over time. They used a frozen-event procedure to convert the dynamic problem into a static one. They developed an integer programming model and presented a heuristic algorithm with the complexity of $O(n^3)$.

Cheng et al. (1999) addressed the $F2,S/ST_{si},R_{si}/C_{max}$ problem for the case where the setup operation is performed by a single server that can perform at most one setup at a time. They addressed the problem under two cases of separable and non-separable setup and removal times. They showed that both cases of the problem are NP-hard in the strong sense. They also proposed some heuristics and analyzed their worst-case error bounds. Glass et al. (2000) proved the NP-hardness of the same problem in the strong sense without removal times, i.e., $F2,S/ST_{si}/C_{max}$. Brucker et al. (2005) addressed the $Fm,S/ST_{si}/C_{max}$ problem with m machines and a single server. They derived complexity results for some special cases and showed that some problems are polynomially solvable. For example, they showed the NP-hardness of the $F2,S/ST_{si}$=s$/C_{max}$, where ST_{si}=s means that the setup times are the same. Brucker et al. (2005) also considered other objective functions including $\sum C_j$, $\sum w_jC_j$, $\sum T_j$, $\sum w_jT_j$, $\sum w_jT_j$, and L_{max}, where they identified some polynomially solvable cases. For example, they showed that the $F2,S/ST_{si}$=s$/\sum w_jT_j$ is polynomially solvable for the case of equal job processing times.

5.1.2 No-wait Flow Shop

A no-wait flow shop problem occurs when the operations of the same job have to be processed contiguously from start to end without interruptions either on or between machines.

Allahverdi and Aldowaisan (2000) considered the $F3/ST_{si},no\text{-}wait/\sum C_j$ problem. They found optimal solutions for problems where the setup and processing times satisfy certain conditions, and established a dominance relation. Furthermore, they presented five heuristics and evaluated the performance of these heuristics through computational experiments. The computational experiments revealed that one of the heuristics has an average error less than 0.01% for up to 18 jobs. The performance of the heuristics was compared with one another for larger number of jobs, up to 100. Aldowaisan and Allahverdi (2004) proposed several heuristics and tested their effectiveness through extensive computational experiments for the $F3/ST_{si},\ R_{si},\ no\text{-}wait/\sum C_j$ problem. They also obtained a dominance relation and presented a lower bound for the problem. Shyu et al. (2004) presented an Ant Colony Optimization algorithm for the $F2/ST_{si}$, $no\text{-}wait/\sum C_j$ problem, and showed that their algorithm outperforms earlier heuristics. Brown et al. (2004) presented non-polynomial time solution methods and a polynomial-time heuristic for the problem Fm/ST_{si}, $no\text{-}wait/\sum f_j$. They also considered the $C_{\max}$ criterion. Since all the jobs are assumed to be ready at time zero, the two criteria of $\sum f_j$ and $\sum C_j$ are equivalent. Ruiz and Allahverdi (2006) presented a dominance relation for the $F4/ST_{si}$,$no\text{-}wait/\sum C_j$ problem, and proposed an iterated local search method for the Fm/ST_{si}, $no\text{-}wait/\sum C_j$ problem. Ruiz and Allahverdi (2006) compared the heuristics of Shyu et al. (2004), Brown et al. (2004), and Allahverdi and Aldowaisan (2000), and showed that one of the heuristics of Allahverdi and Aldowaisan (2000) significantly outperforms the others. Ruiz and Allahverdi (2006) showed that their iterated local search method outperforms the best heuristic of Allahverdi and Aldowaisan (2000).

Dileepan (2004) obtained some dominance relations for the $F2/ST_{si}$, $no\text{-}wait/L_{max}$ problem. Fondrevelle et al. (2005a) considered the same problem but treating the removal times as separated from the processing times, i.e., $F2/ST_{si}$, R_{si}, $no\text{-}wait/L_{max}$. They showed that certain sequences are optimal if certain conditions hold, and proposed a branch-and-bound algorithm that can solve problems with up to 18 jobs. Their computational analysis showed that the branch-and-bound algorithm performs better when the setup and removal times are not too large in comparison with the processing times.

Sidney et al. (2000) studied the $F2/ST_{si}$, $no\text{-}wait/C_{max}$ problem, where the setup time on the second machine consists of two parts. During the first part of the setup, the job must not be present at the machine, while the second part of the setup can be performed in the presence or absence of the job. They proposed a heuristic algorithm, and established its worst-case performance ratio to be 4/3.

Chang et al. (2004b) derived two dominance relations for the $F2/ST_{si}$, R_{si}, $no\text{-}wait/\sum f_j$ problem, where all the jobs are ready at time zero. They also proposed a greedy search heuristic algorithm for the problem. Allahverdi and Aldowaisan (2001) stated that the two-machine problem with the additive sequence-dependent setup times is equivalent to the problem addressed by Chang et al. (2004b). There is no information on comparison of the heuristic proposed by Chang et al. (2004b) and the heuristics presented by Allahverdi and Aldowaisan (2001).

Glass et al. (2000) addressed the $F2,S/ST_{si}$, $no\text{-}wait/C_{max}$ problem for the case where the setup operation is performed by a single server. They reduced the problem to the Gilmore-Gomory traveling salesman problem and solved it in polynomial time.

5.1.3 Flexible (Hybrid) Flow Shop

A flexible flow shop is an extension of a regular flow shop, where in each stage there may be more than one machine in parallel as a result of the need to increase the capacity in that stage. Each job still needs to be processed first in stage 1, then in stage 2, and so on. In some cases, not all the jobs need to go through all the stages but still all the jobs have to follow the same machine route. This problem is known as the flexible flow line problem. It can be assumed that the job has zero processing time in the skipped stage. Therefore, we will refer to both problems as flexible flow shops.

Botta-Genoulaz (2000) studied the $FFm/ST_{si}, R_{si}, prec/L_{max}$ problem with minimum time lags (between two successive operations) such as transportation time. Botta-Genoulaz proposed six different heuristics, and evaluated their performance. Low (2005) addressed the $FFm/ST_{si}, R_{sd} /\sum f_j$ problem, where in each stage there are several unrelated parallel machines and the jobs are ready at time zero. He proposed a heuristic to generate an initial solution, and a simulated annealing algorithm to improve the initial solution. The efficiency of the hybrid approach was tested by computational experiments.

Chang et al. (2004c) addressed the $FF2/ST_{si}, R_{si}, no\text{-}wait/C_{max}$ problem, where there is only one machine in the first stage, while there are *m* parallel machines in the second stage. They developed dominance relations and proposed two heuristics.

Allaoui and Artiba (2004) addressed the FFm/ST_{si} problem with respect to the criteria of C_{max}, T_{max}, $\sum T_j$, $\sum U_j$ and $\sum C_j$ with machine unavailability intervals (due to breakdowns preventive maintenance), where the transportation times between the stages are explicitly considered. Two types of strategies are possible to follow when a job is interrupted as a result of machine unavailability. If the job continues processing after the machine becomes available, the strategy is called preempt-resume, while it is called preempt-restart if the job has to restart processing from the beginning (all prior processing is wasted). Allaoui and Artiba (2004) considered both preempt-resume and preempt-restart strategies. They integrated simulation and optimization to solve the problem. They showed through computational experiments that the performance of their proposed heuristic with respect to the considered criteria is affected by the percentage of repair time.

Logendran et al. (2005a) investigated constructive heuristics for a generalization of the problem $FFm/ST_{si}/C_{max}$, where the jobs are partitioned into several families and jobs of the same family assigned on the same machine should be processed jointly. Machine dependent and sequence (of families) independent setup times are given.

5.1.4 Assembly Flow Shop

In a two-stage assembly flow shop scheduling problem, there are *n* jobs where each job has $k+1$ operations and there are $k+1$ different machines to perform each of these operations. Each machine can process only one job at a time. For each job, the first *k* operations are conducted in the first stage in parallel and a final operation in the second stage. Each of the *k* operations in the first stage is performed by a different machine, and the last operation in the second stage may start only after all the *k* operations in the first stage are completed. The two-stage assembly scheduling problem has many applications in industry. Allahverdi and Al-Anzi (2006b) addressed the $AF2/ST_{si}/C_{max}$ problem. They developed a dominance relation, and proposed three heuristics, including a particle swarm optimization heuristic. Al-Anzi and Allahverdi (2005b) proposed different heuristics, including a self-adaptive differential evolution heuristic, for the $AF2/ST_{si}/L_{max}$ problem, where they compared the heuristics with one another and also with previous heuristics without setup times. They also presented a dominance relation for the problem.

5.1.5. Random Setup Times

Kim and Bobrowski (1997) pointed out that in many real-world situations, setup times may vary as a result of random factors such as crew skills, temporary shortage of equipment, tools and setup crews, and unexpected breakdown of fixtures and tools during a setup operation. They stated that assuming random setup times to be fixed may lead to the development of inefficient results. Moreover, it is sometimes difficult to obtain exact probability distributions of the setup times if modeled as random variables. As such, a solution obtained by assuming a certain probability distribution may not be close to the optimal solution for the realization of the process. It has been observed that although the exact probability distributions of setup times may not be known before scheduling, upper and lower bounds on setup times are easy to obtain in many practical cases. This information on the bounds of setup times is important, and it should be utilized in finding a solution for the scheduling problem.

Realizing this fact, Allahverdi et al. (2003) addressed the $F2/ST_{si}/C_{max}$ problem, where the setup times are random variables with known lower and upper bounds. They established some dominance relations, which help in reducing the set containing an optimal solution for any realization of the setup times. In other words, one of the sequences in the solution set will be optimal regardless of which values the setup times take (some values between the lower and upper bound). In some cases, the set of solutions still could be very large. Allahverdi et al. (2003) assumed that the job processing times are known fixed values. However, in some cases, it may be necessary to model the processing times as random variables with lower and upper bounds, in addition to the setup times. Allahverdi (2005a, b, c) considered the $F2/ST_{si}/C_{max}$, $F2/ST_{si}/\sum C_j$ and $F2/ST_{si}/L_{max}$ problem, respectively, with random and bounded setup and processing times. He obtained some dominance relations to reduce the solution set for each problem.

Allahverdi and Savsar (2001) considered the two-machine flow shop scheduling problem with separate setup times, where the machines are subject to random breakdowns. The setup times become random variables as a result of machine breakdowns. They obtained sequences that minimize the makespan with probability 1, which is also known as almost surely (Pinedo, 1995), when the first or the second machine is subject to random breakdowns without making any assumptions about the distribution of the breakdowns.

5.2. Non-batch sequence-dependent setup times

5.2.1 Flow shop

Rios-Mercado and Bard (1999a, b) addressed the $Fm/ST_{sd}/C_{max}$ problem. Rios-Mercado and Bard (1999a) presented a branch-and-bound algorithm, incorporating lower and upper bounds and dominance elimination criteria, to solve the problem. They provided test results for a wide range of problem instances. Rios-Mercado and Bard (1999b) proposed a heuristic for the same problem, which transforms an instance of the problem into an instance of the traveling salesman problem by introducing a cost function that penalizes both large setup times and bad fitness of a given schedule. Ruiz et al. (2005a) proposed two genetic algorithms for the same problem, and showed that their heuristics outperform that of Rios-Mercado and Bard (1999b) and others. Ruiz and Stützle (2006) presented two simple local search based Iterated Greedy algorithms, and showed that their algorithms perform better than those of Ruiz et al. (2005a). Rios-Mercado and Bard (2003) studied the polyhedral structure of two different mixed-integer programming formulations for the same problem. One is related to the asymmetric traveling salesman problem and the other is derived from an earlier proposed model. The two approaches were evaluated by using a branch-and-cut algorithm, which indicated that the approach related to the asymmetric traveling salesman problem was inferior in terms of the computational time. Stafford and Tseng (2002) also proposed two mixed-integer linear programming models, which are based on the work of Tseng and Stafford (2001), for the same problem. The mixed-integer programming models proposed by Rios-Mercado and Bard (2003) and Stafford and Tseng (2002) were independently developed, and hence, remain to be compared to each other. Tseng et al. (2005) developed a penalty-based heuristic algorithm for the same problem, and compared their heuristic with an existing index heuristic algorithm.

The $Fm/ST_{sd}/C_{max}$ problem was studied by Norman (1999), where there exists buffers with finite capacity between machines. He proposed a tabu search based heuristic and compared it with some other methods. Computational experiments showed the effectiveness of the tabu search approach. Maddux III and Gupta (2003) addressed the $F2/ST_{sd}/C_{max}$ problem with buffers of zero capacity between the machines, and where some jobs leave after the first machine and some jobs continue through the second machine. They developed lower bounds and presented a heuristic to solve the problem.

Hwang and Sun (1997) addressed the problem of a side frame press shop in a truck manufacturing company, where all the jobs need to be processed by two machines. All the jobs require processing by the first machine more than once. Moreover, the setup time required by a job on the first machine depends on the two immediately preceding jobs. Hwang and Sun (1997) redefined the job elements and converted the problem into the $F2/ST_{sd}$, $prec/C_{max}$ problem. They proposed a dynamic programming approach to solve the problem. Hwang and Sun (1998) also considered the same problem, and presented a genetic algorithm to solve the problem. Sun and Hwang (2001) addressed a related problem of $F2/ST_{sd}/C_{max}$, where the setup times are present only on the second machine and the setup time of a job depends on k $(k>1)$ immediately preceding jobs. They proposed a dynamic programming formulation and a genetic algorithm for the problem.

The $Fm/ST_{sd}/\sum w_j f_j$ problem was addressed by Rajendran and Ziegler (1997), where they proposed a heuristic for the problem. They also presented an improvement scheme to enhance the quality of the proposed heuristic. Sonmez and Baykasoglu (1998) developed a dynamic programming formulation for the $Fm/ST_{sd}/\sum w_j T_j$ problem, where they applied the formulation to a plastic pipe manufacturing factory. They reported that an increase in the number of jobs greatly increases the computational time, while an increase in the number of machines has a very small effect on the computational time of the proposed dynamic programming. Rajendran and Ziegler (2003) studied the same problem with a combination of two of the objectives considered by Rajendran and Ziegler (1997) and Sonmez and Baykasoglu (1998), i.e., $Fm/ST_{sd}/\sum w_j f_j + \sum w_j T_j$. Rajendran and Ziegler (2003) proposed heuristics, and compared them with an existing heuristic, a random search procedure, and a greedy local search. Ruiz and Stützle (2006) proposed two simple local search based Iterated Greedy algorithms for the $Fm/ST_{sd}/\sum w_j T_j$ problem. They showed that their algorithms perform better than that of Rajendran and Ziegler (2003) and earlier heuristics.

Andrés et al. (2005b) addressed the Fm/ST_{sd}, $prec$ problem with the objective of minimizing both C_{max} and $n^{-1}\sum T_i$. They proposed a multi-objective genetic algorithm to solve the problem.

For the problems considered so far, a job requires only one operation on a machine. In some cases, e.g., in the semiconductor industry, a job may require to have more than one operation on a machine before the job completes its processing on that machine. These flow shops are known as reentrant flow shops. Demirkol and Uzsoy (2000) addressed the $Fm/ST_{sd},/L_{max}$ problem for a reentrant flow shop. They developed several decomposition methods, and identified an enhanced decomposition method for the problem.

5.2.2 No-wait Flow Shop

Allahverdi and Aldowaisan (2001) considered the $F2/ST_{sd}$, $no\text{-}wait/\sum C_j$ problem. They showed that certain sequences are optimal if certain conditions hold. Moreover, they developed a dominance relation and presented several heuristics with the computational complexities of $O(n^2)$ and $O(n^3)$. The heuristics consist of two phases; in the first phase a starting sequence is developed, and in the second a repeated insertion technique is applied to get a solution. Computational experiments demonstrated that the concept of repeated insertion application is quite useful for any starting sequence, and that the solutions for all the starting sequences converge to about the same value after a few number of iterations.

Bianco et al. (1999) addressed the problem of Fm/ST_{sd}, $no\text{-}wait,\ r_j/C_{max}$. They showed that the problem is equivalent to the asymmetric traveling salesman problem with additional visiting time constraints. They presented lower bounds, an integer programming formulation, and two heuristics for the problem. They also evaluated the performance of the lower bounds and heuristics by using randomly generated data. França et al. (2006) proposed a hybrid genetic algorithm for the same problem. They showed that their hybrid genetic algorithm performs better than the heuristics of Bianco et al. (1999) for a vast majority of randomly generated problem instances. Stafford and Tseng (2002) proposed two mixed-integer linear programming models for the $Fm/ST_{sd},no\text{-}wait/C_{max}$ problem.

5.2.3 Flexible Flow Shop

Liu and Chang (2000) addressed the problem of FFm/ST_{sd}, SC_{sd}, r_j with the objective of minimizing the sum of setup times and costs. They first formulated the problem as a separable integer programming problem. Then Lagrangian relaxation was utilized, and finally a search heuristic was proposed.

Kurz and Askin (2003) studied the $FFm/ST_{sd}/C_{max}$ problem with missing operations, where the machine routes for some jobs contain less than m machines, i.e., all the jobs need not visit all the stages. They explored three types of heuristics for the problem, namely insertion heuristics, Johnson's based heuristics, and greedy heuristics. They identified the range of conditions under which each method performs well. Kurz and Askin (2004) compared four heuristics, including the random keys genetic algorithm, for the problem. They developed lower bounds and utilized these bounds in the evaluation of the proposed heuristics. The computational experiments showed that the random keys genetic algorithm performs best. Zandieh et al. (2006) proposed an immune algorithm, and showed that this algorithm outperforms the random keys genetic algorithm of Kurz and Askin (2004). Logendran et al. (2005b) developed three tabu search-based algorithms for the same problem. In order to aid the search algorithms with a better initial solution, they considered three different initial solution finding mechanisms. A detailed statistical experiment based on the split-plot design was performed to analyze both makespan and computation time as two separate response variables. Ruiz and Maroto (2006) also studied the $FFm/ST_{sd}/C_{max}$ problem, but in a more complex environment, where the machines in each stage are unrelated and some machines are not eligible to perform some jobs. Ceramic tiles are manufactured in such environments. Ruiz and Maroto (2006) proposed a genetic algorithm, where they conducted an extensive calibration of the different parameters and operators by means of experimental designs.

Pugazhendhi et al. (2004) addressed the $FFm/ST_{sd}/\sum w_j f_j$ problem, where some jobs may have missing operations on some machines. They proposed a heuristic procedure to derive a non-permutation schedule from a given permutation sequence. They tested the performance of the proposed heuristics and showed that it performs well.

Jungwattanaki et al. (2005) considered the $FFm/ST_{sd}/\lambda C_{max}+(1\text{-}\lambda)\sum U_j$ problem for the case of unrelated parallel machines, where $0 \leq \lambda \leq 1$. They adapted the well-known constructive heuristics and iterative heuristics (genetic algorithm and simulated annealing) for the pure flow shop problems. The computational results indicated that among the constructive heuristics, a job insertion based heuristic outperforms the other constructive heuristics, while the genetic algorithm outperforms the simulated annealing among the iterative heuristics.

Ruiz et al. (2005b) presented a genetic algorithm for the FFm/ST_{sd}, $prec/C_{max}$ problem, where several realistic characteristics are jointly considered such as release dates for machines, unrelated machines in each stage, machine eligibility, possibility of the setup times to be both anticipatory and non-anticipatory, and time lags.

5.3. Batch sequence-independent setup times

Since not many papers exist in this category, we will not separately analyze them under different shop environments.

Wang and Cheng (2005) derived a number of properties of the optimal solution for the $F2/ST_{si,b}/\sum f_j$ problem, where jobs are ready at time zero. They also proposed a branch-and-bound algorithm and a heuristic to solve the problem. They showed that the branch-and-bound algorithm can solve problems with up to 30 jobs and 10 families. Gupta and Schaller (2006) presented a branch-and-bound algorithm for the same problem with m machines and under the group technology assumption, i.e., $Fm/ST_{si,b}/\sum f_i$. They also proposed and evaluated several heuristics for the problem. Yang and Chern (2000) addressed the $F2/ST_{si,b}$, $R_{si,b}/C_{max}$ problem under the group technology assumption, where a transportation time is required for moving the jobs from the first machine to the second machine. They proposed a polynomial-time algorithm to solve the problem for the case of permutation schedules.

Lin and Cheng (2001) investigated the single family problem $F2/ST_{si,b}$,$no\text{-}wait/C_{max}$ with a common non-anticipatory batch setup time under the batch availability model. The problem was proved to be strongly NP-hard even if all the processing times on one of the machines are equal. A formula for the optimal batch size was given if all the operations of all the jobs have the same processing time. The problem is solvable in $O(n^3)$ time if the job processing times are the same on each machine but different from one machine to the other machine.

Wang and Cheng (2006) addressed the $F2/ST_{si,b}$,$no\text{-}wait/L_{max}$ problem, where they obtained some dominance relations, proposed a branch-and-bound algorithm and a heuristic for the problem. They showed by computational experiments that the branch-and-bound algorithm can solve problems with up to 30 jobs and 10 families, and the heuristic can produce near optimal solutions (on average, the error is less than 1%) for the considered problems.

The reentrant flow shop problem of $F2/ST_{si,b}$ $/C_{max}$ was shown to be NP-hard by Yang et al. (2006) under the assumption of group technology and the assumption that each family consists of identical jobs. They presented some dominance relations and proposed a branch-and-bound algorithm.

Huang and Li (1998) studied the $FF2/ST_{si,b}/C_{max}$ problem under the group technology assumption, where the first stage consists of only one machine and the second stage consists of multiple uniform machines. They presented two heuristics to solve the problem and derived a model to determine the trade-offs between the costs and the speeds of the machines in the second stage.

Several authors have studied a single family flow shop problem under the batch availability model with anticipatory or non-anticipatory machine dependent setup times. For C_{max} minimization, Mosheiov and Oron (2005) suggested an $O(n)$ time algorithm for the m-machine flow shop, equal setup times and equal job processing times. Cheng et al. (2000) considered the above problem under the assumptions that there are two machines, setups are non-anticipatory and equal, and schedules are permutation ones with consistent batches. They proved that this problem is NP-hard in the strong sense, derived several properties of an optimal schedule, and developed $O(n)$, $O(n^2)$ and $O(n^3)$ time algorithms for the cases where (1) all the processing times are the same, (2) the processing times on one of the machines are the same, and (3) the processing times can be oppositely ordered on machines 1 and 2. They also suggested several constructive heuristics for the general case of their problem. Glass et al. (2001) studied the two-machine problem under the assumption that the setups are anticipatory. They proved that when batches are consistent in an optimal schedule, the problem is strongly NP-hard, and derived a heuristic with a tight worst-case performance bound of 4/3. The heuristic constructs a schedule with at most three consistent batches. For the m-machine problems with identical processing time jobs and the criteria of minimizing $\sum C_j$ and C_{max}, Mosheiov et al. (2004) suggested rounding procedures to calculate integer batch sizes from a straightforward solution of the relaxed non-integer batch size problem. Bukchin et al. (2002) and Bukchin and Masin (2004) studied a continuous relaxation of the two- and m-machine problem, respectively, to minimize $\sum C_j$ with machine dependent setup and processing times and consistent batches. Bukchin et al. (2002) suggested a solution method based on solving convex programming problems. The method provides an optimal solution for a certain combination of input parameters. Bukchin and Masin (2004) suggested an enumerative algorithm to construct the Pareto set for the simultaneous minimization of C_{max} and $\sum C_j$.

Cheng and Kovalyov (1998) and Cheng et al. (2004) studied a two-stage problem, in which there is a single machine in the first stage and m machines in the second stage. Each job has to be processed on the first stage machine and then on a specific (dedicated) second stage machine. Thus, the jobs are classified into m families. The job processing time depends solely on its family index. A sequence independent setup time is required on the first stage machine to switch from a job of one family to a job of another family. The objective is to minimize the makespan. Cheng et al. (2004) showed that the problem can be solved in $O(n^m)$ time and it can be solved in $O(n\log L)$ time for the two-machine case, where L is the maximum input parameter. The problem can be used for modeling a disassembly process.

Danneberg et al. (1999) considered a permutation flow shop in which the jobs of different families are processed in consistent batches under the batch availability model. The processing time of a batch is equal to the maximum processing time of its jobs. The setup times are anticipatory, family and machine dependent but sequence (of families) independent.

Batch sizes are bounded from above by the same number. For minimizing C_{max} and Σw_jC_j, Danneberg et al. (1999) suggested constructive heuristics and iterative algorithms, including simulated annealing, tabu search and multilevel search, based on specific neighborhood structures.

A two-machine flow shop problem to minimize the makespan was studied by Pranzo (2004) under the group technology assumption, anticipatory sequence independent setup and removal times and limited intermediate buffer capacity c, $0<c\leq n$. Each job goes through the buffer when it moves from the first to the second machine. If the buffer is full, the job stays on the first machine, which prevents other jobs from being processed on this machine. The job processing times are family and machine dependent. Pranzo derived numbers b_f, $f=1,\ldots,F$, such that if the cardinality of family f exceeds b_f for each family, then the problem reduces to a polynomially solvable traveling salesman problem.

Kovalyov et al. (2004) studied the single family problem $AF2/ST_{si}/C_{max}$ with anticipatory, machine dependent batch setup times under the batch availability model. They proved that the search for an optimal schedule can be limited to permutation schedules, and presented a heuristic with a tight worst case performance bound of $2-1/(m+1)$. The heuristic constructs a schedule with one or two consistent batches.

A two-machine flow shop problem to minimize the makespan was studied by Lin and Cheng (2005) for the case where the first machine processes jobs individually and the second machine in batches under the batch availability model. Each batch is preceded by a constant anticipatory or non-anticipatory setup. The problem was proved to be strongly NP-hard. The case where the processing times on the two machines can be oppositely ordered is solvable in $O(n^2)$ time. Constructive heuristics were presented for the general case.

5.4. Batch sequence-dependent setup times

As in section 5.3, we will not separately analyze the work in this category under different shop environments.

Schaller et al. (2000) developed lower bounds for the $Fm/ST_{sd,b}/C_{max}$ problem under the group technology assumption. They also proposed a heuristic to solve the problem, and empirically evaluated the performance of the proposed heuristic. França et al. (2005) proposed a genetic, a memetic, and a multi-start algorithm for the same problem. They showed that all of the three algorithms outperform the heuristic proposed by Schaller et al. (2000). They also concluded that the mimetic algorithm slightly outperforms the genetic and multi-start algorithms. Logendran et al. (2006) also considered the same problem for the two-machine case, and developed three search algorithms based on tabu search. They developed lower bounds and used these bounds in the evaluation of the developed algorithms. Clearly, the search algorithms of Logendran et al. (2006) and the memetic algorithm of França et al. (2005) remain to be compared. Cho and Ahn (2003) considered the same model with the criterion of minimizing the total tardiness, and suggested a hybrid genetic algorithm in which a genetic algorithm was used to determine the group sequence, while a heuristic procedure was used to determine the job sequence in each group.

Lin and Liao (2003) considered a scheduling problem from a label sticker manufacturing company, and stated that it is equivalent to a generalization of the $FF2/ST_{sd,b}/T_{max}$ problem, where the first stage consists of a single high speed machine and the setup times exist only on the first machine. Since each job has a weight, the objective is to minimize the weighted maximum tardiness. They proposed a heuristic, and showed that it outperforms the current practice in the company. Andrés et al. (2005a) studied the problem of scheduling in a tile company. They modeled the problem as $FF3/ST_{sd,b}/C_{max}$. Their main goal was to identify a set of families with common features. They proposed a heuristic using their defined "coefficient of similarity" between the jobs and successfully applied it in the tile company.

Hall et al. (2003) studied the problem $F/ST_{sd,b}$, *no-wait*/C_{max} with family and machine dependent setup and processing times under the batch availability model. Setups are either all anticipatory or all non-anticipatory. Jobs of the same family, though can be partitioned into batches, are required to be processed consecutively on each machine. A setup occurs only between batches of different families. The problem reduces to a generalized traveling salesman problem. A customized heuristic was proposed. A pseudopolynomial-time algorithm was presented for the single family case.

Reddy and Narendran (2003) studied a five-machine permutation flow shop problem with dynamically arriving jobs belonging to different families in a stochastic environment, where both the processing times and the time between arrivals are assumed to be exponential random variables. Sequence dependent setup times occur between the jobs of distinct families. Reddy and Narendran compared the quality of nine heuristics (a combination of three dispatching rules and three queue selection rules) with regard to simulated data. Their objective was to minimize (i) the average job time in the system, (ii) the average job tardiness, and (iii) the percentage of tardy jobs.

6. Job shop and open shop

A job shop environment consists of *m* different machines and each job has a given machine route in which some machines can be missing and some can repeat. On the other hand, in an open shop, each job should be processed once on each of the *m* machines passing them in any order.

Cheung and Zhou (2001) proposed a hybrid genetic algorithm, based on a genetic algorithm and heuristic rules, for the problem of $J/ST_{sd}/C_{max}$. They showed by computational analysis that their hybrid algorithm is superior to earlier methods proposed for the same problem. Ballicu et al. (2002) considered the same problem and represented it in terms of disjunctive graphs. They also derived a mixed integer linear programming model. Choi and Choi (2002) presented another mixed integer programming model for the same problem and a local search scheme. The local search scheme utilizes a property that reduces computational time. By using benchmark data, Choi and Choi (2002) showed that the scheme significantly enhances the performance of several greedy-based dispatching rules. A fast tabu search heuristic was proposed by Artigues and Buscaylet (2003) for the problem. Artigues et al. (2005a) obtained upper bounds by a priority rule-based multi-pass heuristic. A branch-and-bound procedure was proposed by Artigues et al. (2004), who improved the results obtained by the branch-and-bound procedure presented by Focacci et al. (2000). Artigues et al. (2005b) presented a synthesis of the methods described by Artigues and Buscaylet (2003), Artigues et al. (2004), and Artigues et al. (2005a) by integrating tabu search with multi-pass sampling heuristics. The heuristics proposed by Cheung and Zhou (2001) and Artigues et al. (2005b) remain to be compared.

Sun and Yee (2003) addressed the $J/ST_{sd}/C_{max}$ problem but with the additional characteristic of reentrant work flows. They utilized disjunctive graph representation of the problem, and proposed several heuristics including a genetic algorithm. Balas et al. (2005) formulated the $J/ST_{sd}, r_j/C_{max}$ problem as an asymmetric traveling salesman problem with a special type of precedence constraints, which can be solved by a dynamic programming algorithm whose complexity is linear in the number of operations.

The $J/ST_{sd}/L_{max}$ problem was addressed by Artigues and Roubellat (2002), where they proposed a polynomial insertion algorithm to solve the problem. Sun and Noble (1999) decomposed the $J/ST_{sd}, r_j/\sum w_jT_j^2$ problem into a series of single-machine scheduling problems within a shifting bottleneck framework. They solved the problem using a Lagrangian relaxation based approach. The $J/ST_{sd}, r_j, prec/\sum T_j$ problem was considered by Tahar et al. (2005) for the case of hybrid job shop (with identical parallel machines in some stages), where precedence constraints exist between some jobs. They proposed an Ant Colony Algorithm and showed by computational analysis that it performs better than a genetic algorithm.

Sotskov et al. (1999) considered the $J/ST_{si,b}$ problem with respect to both regular and non-regular criteria. They proposed different insertion techniques combined with beam search to solve the problem. The insertion techniques were tested on a large collection of test problems and compared with other constructive algorithms based on priority rules.

Zoghby et al. (2005) investigated the feasibility conditions for metaheuristic searches for the case of reentrancy in the disjunctive graph model of the job shop scheduling problem, where the setup times are considered as sequence dependent. They presented the conditions under which infeasible solutions occur, and proposed an algorithm to remove such infeasibilities.

A job shop problem with families of identical jobs and sequence (of families) independent, machine dependent anticipatory setups to minimize C_{max} was studied by Low et al. (2004). They gave a disjunctive graph presentation of the problem, and suggested an integer programming algorithm.

A problem in the reentrant job shop environment was considered by Aldakhilallah and

Ramesh (2001), where a batch of identical jobs has to be repeatedly processed in a job shop according to a given machine sequence. A machine dependent setup time precedes each batch processing on each machine. The objective is to find a batch size and a (cyclic) schedule such that the flowtime of the batch and the length of one cycle are simultaneously minimized. Two constructive heuristics were proposed.

A Petri net approach was suggested by Artigues and Roubellat (2001) for on-line and off-line scheduling of a job shop with job release dates, sequence dependent family setup times and the maximum lateness objective. A set of solutions to a static scheduling problem represented by an acyclic directed graph was assumed to be predetermined, as input of their proposed decision support system for the considered scheduling problem.

Glass et al. (2000) showed that the $O2,S/ST_{si}/C_{max}$ problem is NP-hard in the strong sense when the setup operations are performed by a single server. Glass et al. (2001) showed that the $O2/ST_{si,b}/C_{max}$ problem is NP-hard in the ordinary sense. Strusevich (2000) studied the same problem and proposed a linear time heuristic algorithm. He showed that the algorithm can guarantee a worst-case performance ratio less than 5/4.

Averbakh et al. (2005) studied a problem that is equivalent to $O2/ST_{sd,b}/C_{max}$ with two families. They proved that the optimal makespan value falls in the interval $[C,(6/5)C]$, where C is a trivially calculated lower bound, and suggested an $O(n)$ time algorithm to construct a schedule with makespan from this interval.

Blazewicz and Kovalyov (2002) proved the ordinary NP-hardness of the problem $O2/ST_{si,b}/\sum C_j$ under the group technology assumption, and showed that omitting this assumption does not lead to an equivalent problem.

A single family open shop problem with equal job processing times and a common non-anticipatory setup time to minimize C_{max} or $\sum C_j$ under the batch availability model was studied by Mosheiov and Oron (2006b). They suggested a constant time solution for C_{max} minimization, and an $O(n)$ time heuristic for $\sum C_i$ minimization.

7. Others

Baki and Vickson (2003, 2004) and Cheng and Kovalyov (2003) studied a two-machine flow shop problem in which the processing of the jobs requires the continuous presence of a single operator. The operator can serve one machine at a time and there is a machine dependent setup time when it switches to a particular machine. Baki and Vickson (2003) derived an $O(nlogn)$ time algorithm to minimize L_{max}. Baki and Vickson (2004) and Cheng and Kovalyov (2003) proved the NP-hardness of minimizing $\sum U_j$, and suggested pseudopolynomial-time algorithms for minimizing $\sum w_jU_j$. Cheng and Kovalyov (2003) showed a relationship of the flow shop problem and a single machine single family batch scheduling problem, which leads to the following results: $O(nlogn)$ time algorithms for minimizing L_{max} and for minimizing $\sum C_j$ in the case of agreeable processing times, strong NP-hardness of minimizing $\sum w_jC_j$, and dynamic programming algorithms for minimizing

$\sum w_j U_j$.

Baki and Vickson (2003) also studied a single operator problem in the two-machine open shop environment. The problem is equivalent to the problem $1/ST_{si,b}/L_{max}$ with two families, and can be solved in $O(n\log n)$ time by the algorithm of Wagelmans and Gerodimos (2000).

Iravani and Teo (2005) considered an m-machine flow shop in which the processing times are machine dependent, and there are machine dependent setup costs and holding costs for a job to stay one time unit on a machine. The objective is to minimize the total setup and holding cost. They introduced a so-called chain-like structure schedule, and proved that such a schedule is asymptotically optimal as $n \to \infty$. They also proved that any algorithm in a natural class of algorithms is a 2-approximation algorithm for the considered problem.

Valls et al. (1998) addressed the problem of a machine workshop in a Spanish company that produces heavy boat engines and electricity power stations. The problem is a generalization of the job shop scheduling problem, where in some stations there is more than one machine and the machines need to be set up if the coming job is in a different family. Each job has a release date and a due date. The objective is to minimize the makespan. Valls et al. (1998) proposed a tabu search algorithm to solve the problem.

Leu (1999) studied cellular flexible assembly systems that produce low-volume, large products in an assemble-to-order environment such as the assembly of weapons and heavy machinery. Leu proposed two heuristics, namely a single-stage and two-stage heuristics. The two-stage heuristic attempts to serially process similar orders and eliminate the major setup times required between different families. The two-stage heuristic was statistically shown to perform better than the single-stage heuristic with respect to different measures. Leu and Wang (2000) studied the same problem in a hybrid order shipment environment. They proposed a single-stage heuristic and a two-stage heuristic, and showed by computational experiments that the two-stage heuristic outperforms the single-stage heuristic.

Norman and Bean (1999) investigated a scheduling problem that arises from an automaker. They stated that there are many factors that complicate such a problem. Among them, there are different job release and due dates that range throughout the study of the horizon. Moreover, each job has a particular tooling requirement, and tooling conflicts may arise since there is only one copy of each tool and several jobs may require the same tool. Also, the setup times are sequence dependent and the machines are not identical. Norman and Bean (1999) proposed a genetic algorithm for the problem, proved its convergence and tested its performance on data sets obtained from the auto industry.

Pearn et al. (2002a, 2002b) and Yang et al. (2002) considered a scheduling problem in wafer probing factories. In these factories, jobs are clustered by their product types and must be completed before their due dates. The setup times are sequence dependent. Pearn et al. (2002a, 2002b) and Yang et al. (2002) considered the problem to find a schedule that minimizes the total setup time. Pearn et al. (2002a) and Yang et al. (2002) formulated the problem as an integer programming problem, and Pearn et al. (2002b) transformed it into the vehicle routing problem with time windows. Pearn et al. (2002b) also presented three heuristic algorithms with job insertions for the problem. Pearn et al. (2004a) considered the same problem. They computationally tested four existing so-called "saving" vehicle-routing heuristics and three new modifications. The tests demonstrated high efficiency of the modified heuristics. The idea of a saving heuristic is to insert pairs of jobs based on their setup time characteristic called saving. Saving of a pair of jobs includes three setup times: two setup times assuming that the jobs are the first and the last jobs in the schedule and a setup time between the jobs of the pair. Pearn et al. (2004b) addressed the integrated-circuit final testing scheduling problem with reentry of some operations with the same objective function of minimizing the total machine workload. They presented three fast network heuristic algorithms for solving the problem. Ellis et al. (2004) addressed the wafer test scheduling problem with the objective of minimizing makespan. They stated that the jobs have precedence constraints because the test processes are conducted in a specified order on a wafer lot, and setup times are sequence-dependent. They proposed four heuristics for the problem, and applied the heuristics to actual data from a semiconductor manufacturing facility. The results showed that the makespan is reduced by 23-45%. Crama et al. (2002) surveyed the literature on the printed circuit board assembly planning problems with an emphasis on the classification of the related mathematical models. The general models they considered include the simultaneous production sequencing with family setups, an assignment of setup facilities, and determination of setup policies.

Mason et al. (2002) addressed a scheduling problem in semiconductor wafer fabrication facility, where the facility is described as a complex job shop with reentrant flow of products, batching machines, and sequence-dependent setup times. They proposed a modified shifting bottleneck heuristic for the problem to minimize the total weighted tardiness.

Van Hop and Nagarur (2004) considered the printed circuit boards (PCB) problem. They stated that the problem has three sub-problems, which are (i) classifying the PCBs into m groups, where m is the number of available machines, (ii) finding the sequence for each machine, and (iii) component switching, which includes the setup operations. They proposed a genetic algorithm to solve the problem with the objective of minimizing the makespan, which was shown to be the same as minimizing the maximum number of component switches. Leon and Peters (1998) proposed and evaluated a number of strategies for operating a single printed circuit board assembly machine. The strategy adapting the group technology principle was shown to be applicable when component commonality is high and changeovers are time consuming. Partial setup strategies, which allow product families to be split and focus on minimizing the total production time, are shown to adapt to changing production conditions and therefore outperform the other setup strategies.

Chang et al. (2003) considered a biaxially oriented polypropylene (BOPP) film factory, which produces products such as adhesive tapes, photo albums, foodstuff packages, book covers, etc., where the job setup times are sequence-dependent. They proposed a genetic algorithm with variable mutation rates for the problem, and showed that the algorithm outperforms the current practice in the factory.

Queues of tow/barges form when a river lock is rendered inoperable due to several reasons including lock malfunction, a tow/barge accident, and adverse lock operating conditions. Nauss (2006) developed model formulations that allow queues of tow/barges to be cleared using a number of differing objectives in the presence of different setup times between successive passages of tow/barges through the lock. He presented linear and nonlinear integer programming formulations, and carried out computational experiments on a representative set of the problems, showing that the solution approaches generate improved solutions over the current practice.

Aubry et al. (2006) addressed the problem of minimizing the setup costs of a workshop modeled with parallel multi-purpose machines by ensuring that a load-balanced schedule exists. They showed that the problem is NP-hard in the strong sense, and presented the problem as a mixed integer linear program. They also showed that the problem in certain cases can be stated as a transportation problem.

Monkman et al. (2006) proposed a heuristic for a production scheduling problem at a high volume assemble-to-order electronics manufacturer. The proposed heuristic involves assignment, sequencing, and time scheduling steps, with an optimization approach developed for each step. They compared the setup costs resulting from the use of the proposed heuristic against a heuristic previously developed and implemented at the electronics manufacturer. A reduction of setup costs, about 20%, was achieved by applying the proposed heuristic.

Havill and Mao (2006) studied the problem of scheduling online perfectly malleable parallel jobs with arbitrary times on two or more machines. They took into account the setup time to create, dispatch, and destroy multiple processes. They presented an algorithm to minimize makespan.

Yokoyama (2006) described a scheduling model for a production system including machining, setup and assembly operations. Production of a number of single-item products is ordered. Each product is made by assembling a set of several different parts. First, the parts are produced in a flow shop consisting of m machines. Then, they are assembled into products on a single assembly stage. Setup time is needed when a machine starts processing the parts or it changes items. The objective function is the mean completion time for all the products. Yokoyama (2006) proposed solution procedures using pseudo-dynamic programming and a branch-and-bound algorithm.

Mika et al. (2006) addressed the multi-mode resource-constrained project scheduling problem with schedule-dependent setup times. A schedule-dependent setup time is defined as a setup time dependent on the assignment of resources to activities over time, when resources are, e.g., placed in different locations. In such a case, the time necessary to prepare the required resource for processing an activity depends not only on the sequence of activities but, more generally, on the locations in which successive activities are executed. They proposed a tabu search heuristic to solve the problem.

8. Conclusions

We have surveyed more than 300 papers on scheduling with setup times (costs) that have appeared since 1999. On average, more than 40 papers per year have been added to the related literature. As compared to the 190 papers surveyed by Allahverdi et al. (1999) in over 25 years (i.e., about 8 papers per year), there has been a huge jump in the annual research output on scheduling with setup times (costs) in the past six years.

This survey classifies the literature on setup times (costs) according to (1) shop environments, including single machine, parallel machines, flow shops, job shops, open shops, and others; (2) batch and non-batch setup times (costs); (3) sequence-dependent and sequence-independent setup times (costs); and (4) job and batch availability models. The research status on different problem types is reviewed and summarized in Tables 1 – 8. Single machine, parallel-machine, flow shop, job shop and open shop problems were addressed in about 80, 70, 100, 20 and 10 papers, respectively. For single machine problems, three quarters of the papers considered batch setup times while only one quarter of the papers discussed non-batch setup times. For other shop environments, this trend was not the case. For example, for the parallel-machine case, the majority of the papers considered non-batch setup times. Moreover, two thirds of the papers on flow shop problems addressed the non-batch setup times. The majority of the papers addressed sequence-independent setup times because dealing with sequence-dependent setup times is more difficult.

The common solution methods are branch-and-bound algorithms, mathematical programming formulations, dynamic programming algorithms, heuristics and meta-heuristics. Among metaheuristic methods, genetic algorithms were used in about 35 papers while tabu search was used in about half of this number of papers. Simulated annealing was also used in several papers but less than tabu search. Few papers utilized ant colony while particle swarm optimization (PSO) was only used in one paper. The performance of these heuristics, to some extent, depends on different parameters and the operators used as well as on the characteristics and size of problem instances. In some cases, the use of local search methods, while in some other cases, that of hybrid meta-heuristics shows better results. This indicates that different methods have their strengths and weaknesses.

In Table 9, we present the problems for which the computational complexity was reported as unknown in the latest literature. Problems open with respect to strong NP-hardness are marked with an asterisk. We assume a reasonable encoding scheme (see Garey and Johnson, 1979) for each problem, i.e., if the problem formulation explicitly states that there are k parameters equal to a, all these parameters are encoded with two numbers k and a. The most vexing open problem is $1/ST_{si,b}/\sum Ci$.

Table 9. Open problems.

Problem description	**Additional characteristics**	**Reference**

Job availability		
$1/ST_{si,b}/\sum(w_j)C_j$		Potts and Kovalyov (2000)
$Pm/ST_{si,b}/\sum C_j$	m is constant	Potts and Kovalyov (2000)
$P/ST_{si,b},\ p_j=p,\ C_j\leq d_j/-$	Equal setup times s, s is not a multiple of p	Brucker et al. (1998)
$Q_m/ST_{si,b},\ p_j=p,\ C_j\leq d_j/-$	Equal setup times s, s is not a multiple of p, constant m	Brucker et al. (1998)
$F2/ST_{si,b}/C_{max}$	Machine independent setup times	Kleinau (1993)
$^*F2/ST_{si,b}/C_{max}$		Kleinau (1993)
$O2/ST_{si,b}/C_{max}$	Machine independent setup times	Kleinau (1993)
$^*O2/ST_{si,b}/C_{max}$		Kleinau (1993)
$^*O2/ST_{si,b}/C_{max}$	Group technology assumption	Blazewicz and Kovalyov (2002)
Batch availability		
$1/ST_{si,b}/\sum(w_j)C_j$		Cheng et al. (1994)
$1/ST_{si,b}/\sum w_jC_j$	One family, batch setup and processing times are equal to the maximum of job setup and processing times in the batch	Dang and Kang (2004)
$1/ST_{si,b}/\sum C_j$	One family, resource dependent processing times	Ng et al. (2003)
$1/ST_{si,b}/L_{max}$	One family, bounded batch sizes	Cheng and Kovalyov (2001)
$1/ST_{si,b}/\sum U_j$	One family, bounded batch sizes	Cheng and Kovalyov (2001)
$1/ST_{si,b},\ p_j=p/\sum C_j$	One family, bounded batch sizes	Cheng and Kovalyov (2001)
$P/ST_{si,b}/\sum C_j$	One family	Cheng et al. (1996)
Single server problems		
$Pm,S/ST_{si,b}/\sum U_j$	Constant $m\geq 4$	Brucker et al. (2002)
$P,S/ST_{si,b},\ p_j=p,\ r_j/L_{max}$	Unit setup times	Brucker et al. (2002)
$P,S/ST_{si,b},p_j=p,\ r_j/\sum w_jC_j$	Unit setup times	Brucker et al. (2002)
$P,S/ST_{si,b},p_j=p,\ r_j/\sum w_jU_j$	Equal setup times	Brucker et al. (2002)
$P,S/ST_{si,b},p_j=p,\ r_j/\sum w_jT_j$	Equal setup times	Brucker et al. (2002)

Some classes of problems and solution methods have received less attention of the research community than the others. In Table 10, we enumerate some of these classes and give plausible reasons for their being less studied.

Table 10. Less studied classes of problems and methods.

Problems	***Reasons for limited studies***
Problems with setup costs	Time reduction usually implies cost reduction
Multi-machine problems	Hardness, simplifying to a single bottleneck machine
Multi-criteria problems	More difficult than single criterion
Problems with multiple families under the batch availability model	Novelty of the model, unreported applications
Problems with bounded batch sizes	Complicated structure of an optimal solution
Stochastic problems	More difficult than deterministic counter parts
Methods	***Reasons for limited studies***
Heuristics with performance guarantees	Absence of good lower bounds
On-line algorithms	Bad competitive ratio in most cases
Ant colony metaheuristic	Good performance in rare cases
Particle swarm optimization metaheuristic	Good performance in rare cases

New trends in scheduling with setup times or costs include investigation of problems with resource-dependent job and setup parameters, job and setup deterioration, and job or batch transportation. Corresponding applications of such models are found in supply chain management and logistics.

For flow shop, job shop, and open shop problems, the vast majority of the surveyed papers addressed completion time based performance measures (C_{max}, ΣC_j). Therefore, future research on these problems should be more focused on due date related performance measures (L_{max}, T_{max}, ΣT_j, ΣU_j).

Only few papers addressed multi-criteria scheduling problems with setup times. Since most practical problems involve both setup times and multiple objectives, future research on scheduling problems with setup times to optimize multiple objectives is both desirable and interesting.

Stochastic scheduling problems, where some characteristics of the job are modeled as random variables and/or machines may be subject to random breakdowns, with separate setup times, have been addressed only in a few papers. Therefore, another worthy direction of research is to address stochastic scheduling problems with separate setup times.

The number of case studies has considerably increased over the last several years. Most of them are limited to planning activities in manufacturing. However, we believe that setup/cost scheduling models have great potential to be applied in such areas as logistics, telecommunications, electronic auctions and trade, and high-speed parallel computations

Our final conclusion is that if the number of publications on scheduling with setup times or costs continues to grow at the present pace, which we expect to be the case as scheduling research in this area is a fertile field for future research, we suggest that future surveys in this area be devoted to either particular classes of these problems, e.g., based on shop environment, or be focused on specific solution methods.

Acknowledgements

This research was initiated when Prof. Allahverdi was invited to visit Hong Kong. The research was supported in part by The Hong Kong Polytechnic University under grant number A628 from the *Area of Strategic Development in China Business Services*. Prof. Kovalyov was partially supported by INTAS under grant number 03-51-5501. We would like to thank three reviewers for their constructive comments and suggestions that have significantly improved the presentation of the paper. We express our special thanks to Dr Yakov Shafransky for his deep consideration of the definitions we used and a critical analysis of our original presentation.

Table 1. Single machine non-batch setup time scheduling problems

Setup type	References	Criterion (Comments)	Approach/Result
ST_{si} / SC_{si}	Kuik and Tielemans (1997)	Effect of batch sizes on setup utilization (total setup time divided by total setup and processing time) in a queuing delay batching model	An upper bound of 3-2√2 on optimal setup utilization, batch sizes should be corrected if setup utilization is higher than the bound
	Graves and Lee (1999)	L_{max}, $\sum w_j C_j$ (maintenance is performed on the machine periodically)	NP-complete of the problem, dynamic programming
	Liu and Cheng (2002)	Meeting deadlines (r_j, preemption)	NP-hard in the strong sense, dynamic programming, an approximation scheme
	Liu and Cheng (2004)	$\sum C_j$ (r_j, constant setup time, preemption)	NP-hard in the strong sense, worst-case performance ratio
ST_{sd}/ SC_{sd}	Tan and Narasimhan (1997)	$\sum T_j$	Simulated annealing
	Wang and Wang (1997)	Minimize a penalty function including penalties of early-tardy and total setup time	Hybrid genetic algorithm
	Kolahan and Liang (1998)	$\sum w_j E_j + \sum w_j T_j +$(linear costs for compression or extension of job processing times)	Tabu search
	Asano and Ohta (1999)	T_{max} (r_j, machine unavailability for certain periods)	Branch-and-bound, algorithm
	Miller et al. (1999)	Minimize sum of setup costs	A hybrid genetic algorithm
	Armentano and Mazzini (2000)	$\sum T_j$	Genetic algorithm, integer programming
	Tan et al. (2000)	$\sum T_j$	Branch-and-bound, simulated annealing, genetic algorithm, pairwise interchange
	França et al. (2001)	$\sum T_j$	Memetic algorithm, genetic algorithm
	Gagne et al. (2002)	$\sum T_j$	Ant colony
	Mendes et al. (2002a)	$\sum T_j$	Multi start procedure
	Shin et al. (2002)	$L_{max}(r_j)$	Tabu search
	Chang et al. (2004a)	$\sum w_j T_j$ (r_j)	Mathematical programming, heuristic
	Lee and Asllani (2004)	C_{max} and $\sum U_j$ (dual criteria)	Mixed-integer programming, genetic algorithm
	Rabadi et al. (2004)	$\sum E_j + \sum T_j$ (common due date)	Branch-and-bound algorithm
	Eren and Guner (2006)	$\lambda \sum C_j + (1-\lambda) \sum T_j$	Integer programming, Tabu search
	Gupta and Smith (2006)	$\sum T_j$	A greedy randomized adaptive search procedure, a space-based local search heuristic
	Koulamas and Kyparisis (2006)	C_{max}, $\sum C_j$, the total absolute differences in completion times (setup times are proportionate to the length of the already scheduled jobs)	A sorting procedure

Table 2. Single machine batch setup time scheduling problems

Setup type	References	Criterion (Comments)	Approach/Result
$ST_{si,b}$ / $SC_{si,b}$	Azizoglu and Webster (1997)	$\Sigma w_jE_j + \Sigma w_jT_j$ (common due date)	Branch-and-bound, beam search procedure
	Chen et al. (1997)	Find an optimal common due date, and minimize the sum of the costs of tardy jobs (also considers the case of group technology)	Conditions for the optimality, polynomial time algorithm
	Liaee and Emmons (1997)	ΣU_j (Group technology)	NP-hard
	Pan and Su (1997)	L_{max}	Lower bound, branch-and-bound
	Pan and Wu (1998)	Σf_j (subject to due date constraints, group technology)	Presents a polynomial time algorithm
	Webster et al. (1998)	$\Sigma w_jE_j + \Sigma w_jT_j$ (common due date)	Genetic algorithm
	Woeginger (1998)	D_{max}	Dynamic programming
	Yang and Liao (1998)	ΣC_i (a job is attributed a family and an order)	Branch-and-bound algorithm
	Baker (1999)	L_{max} (common setup time)	Heuristic
	Liu and Yu (1999)	ΣU_j (Group technology)	NP-hard in the strong sense (even for the unit processing time and zero setup times)
	Van Oyen et al. (1999)	Σw_jf_j, Σw_jT_j, *and* L_{max} *in expected sense (*processing times and due dates are random variables, consider also group technology*)*	Derives conditions under which the deterministic results are optimal
	Baker and Magazine (2000)	L_{max}	Dominance properties, branch-and-bound algorithm
	Dunstall et al. (2000)	Σw_jf_j	Branch-and-bound algorithm
	Cheng et al. (2001a)	ΣU_j	Strongly NP-hard
	Pan et al. (2001)	L_{max}	Heuristic, integer programming
	Liao and Liao (2002)	Σf_j *(*major and minor setup times*)*	Tabu search
	Shufeng and Yiren (2002)	L_{max} (Group technology, major and minor setup times)	Integer programming, tabu search based heuristic
	Wang and Zou (2002)	L_{max} (major and minor setup times)	Mixed-integer programming, tabu search-based heuristic
	Cheng et al. (2003a)	L_{max}	Strongly NP-hard
	Suriyaarachchi and Wirth (2004)	$\Sigma w_jE_j + \Sigma w_jT_j$ (common due date)	Necessary optimal conditions, heuristic, genetic algorithm
	Schaller (2006)	ΣT_j (consider also group technology)	Branch-and-bound, heuristic
	Schaller and Gupta (2006)	$\Sigma E_j + \Sigma T_j$ (consider also group technology)	Branch-and-bound, heuristic
	Yang and Chand (2006)	ΣC_j (processing times change based on their positions in a schedule)	Lower bound, branch-and-bound
$ST_{sd,b}$/ $SC_{sd,b}$	Van Der Veen (1998)	C_{max}	Traveling salesman problem, polynomial time algorithm
	Sun et al. (1999)	$\Sigma w_jT_j^2$ (r_j)	Lagrangian relaxation based approach, tabu search, simulated annealing
	Dang and Kang (2004)	Σw_jC_j (batch setup and processing times are equal to the maxima of job setup and processing times, respectively, in the batch)	2-approximation, computational complexity is open
	Baptiste and Le Pape (2005)	Minimize a regular sum of objective functions (r_j, setup times and costs)	Lower bounds, dominance properties, branch-and-bound
	Gupta and Sivakumar (2005)	Minimize average tardiness and cycle time, maximize machine utilization	Pareto optimal solution, simulation
	Karabati and Akkan (2006)	ΣC_j	Branch-and-bound
	Sourd (2005)	Minimize the sum of earliness-tardiness and setup costs	Branch-and-bound, dominance rules, heuristic
$ST_{si,b}$, batch availability	Hochbaum and Landy (1997)	$\Sigma w_j C_j$ (common setup times, common batch sizes, p_i=p, two distinct weights)	An algorithm with the time complexity of $O(\sqrt{n} \log n)$
	Baptiste (2000)	Σw_jU_j, Σw_jC_j , ΣT_j $(r_j, p_j=p)$	$O(n^{14})$, dynamic programming
	Baptiste (2000)	T_{max} $(r_j, p_j=p)$	$O(n^{14} log n)$, dynamic programming
	Gerodimos et al. (2000)	ΣU_i (standard and specific components, batching for standard components)	NP-hard, $O(n^2 d_{max})$, d_{max} is the maximum due date, dynamic programming

	Wagelmans and Gerodimos (2000)	L_{max} (standard and specific components, batching for standard components)	*O(nlogn),* dynamic programming
	Baptiste and Jouglet (2001)	ΣT_j	$O(n^{11}M^7)$ dynamic programming, *M* is maximum numerical parameter
	Cheng et al. (2001b)	L_{max} (an upper bound on total weighted resource consumption), and total weighted resource consumption (job deadlines)	Polynomial algorithms, linear programming with two variables
	Cheng and Kovalyov (2001)	Various regular objectives (constant number *k* of distinct job processing times or due dates)	Algorithms polynomial in *n* and exponential in *k*, dynamic programming
	Ng et al. (2002a)	L_{max} (precedence constraints)	$O(n^2)$
	Ng et al. (2003a)	$\Sigma_j C_j$ (an upper bound on total weighted resource consumption), and an inverse problem. Job and batch dependent resources. Agreeable job processing time bounds.	Polynomial algorithms
	Ng et al. (2003b)	ΣC_j (precedence constraints, r_j, $p_j=p$)	$O(n^5)$
	Agnetis et al. (2004)	*TSC* (tool loading)	Heuristic
	Ng et al. (2004)	$\Sigma_j C_j$ (an upper bound on total weighted resource consumption) and an inverse problem	Polynomial algorithms, linear programming with two variables
	Yang (2004b)	ΣC_j (standard and specific components, batching for standard components)	Branch-and-bound algorithm
	Yuan et al. (2004)	L_{max} (precedence constraints, r_j, $p_j=p$)	$O(n^2)$ time reduction to the problem without precedence constraints
	Mosheiov et al. (2005)	ΣC_j ($p_j=p$)	*O(n),* rounding optimal non-integer batch sizes
	Yuan et al. (2005b)	$\Sigma w_j C_j$ (common setup times, common batch sizes)	Strongly NP-hard even if the batch size = 3 and the weight of each job = its processing times
	Mosheiov and Oron (2006a)	ΣC_j ($p_j=p$, bounded batch sizes)	*O(n),* rounding optimal non-integer batch sizes
$ST_{si,b}$, job availability	Gerodimos et al. (2001)	ΣU_j (standard and specific components, batching for standard components)	NP-hard, $O(n^2 d_{max})$, dynamic programming
	Gerodimos et al. (2001)	L_{max} (standard and specific components, batching for standard components)	$O(n^2)$, dynamic programming
	Gerodimos et al. (2001)	ΣC_j (standard and specific components, batching for standard components, agreeable processing times)	*O(nlogn)*, dynamic programming
	Yuan et al. (2005a)	C_{max}, (r_j)	Strongly NP-hard even when the job processing times are unit and the setup times of the families are identical.
$ST_{si,b}$, multi-operation jobs	Gerodimos et al. (1999)	L_{max},	Strongly NP-hard, equivalent to $1/ST_{sib}/L_{max}$
	Gerodimos et al. (1999)	ΣU_j	Strongly NP-hard, pseudopolynomially solvable for fixed number of operations
	Ng et al. (2002b)	ΣC_j	Strongly NP-hard
	Cheng et al. (2003b)	ΣU_j	Strongly NP-hard even when the due-dates are common and all jobs have the same processing time
$ST_{si,b}$	Janiak et al. (2005)	Total weighted resource consumption (job deadline, group technology)	Polynomial algorithms, geometric techniques
	Ng et al. (2005)	$\Sigma w_j C_j$ (an upper bound on total weighted resource consumption, group technology)	Polynomial algorithms, linear programming with two variables
$ST_{si,b}$	Soric (2000a,2000b)	Average work backlog (dynamically arriving jobs)	On-line heuristic, MILP formulation, cutting plane branch-and-bound algorithm
	Vieira et al. (2000)	Average flowtime, machine utilization, setup frequency, rescheduling frequency (stochastic environment with machine breakdowns, dynamically arriving jobs)	Rescheduling algorithm

Table 3. Parallel machines non-batch setup time scheduling problems

Setup type	References	Criterion (Comments)	Approach/Result
ST_{si} / SC_{si}	Koulamas (1996)	Minimizing machine idle time resulting from unavailability of the server (setup is performed by a single server, only two machines)	NP-hard in the strong sense, beam search heuristic
	Kravchenko and Werner (1997)	C_{max}, minimizing the amount of time in list scheduling when some machine is idle due to the unavailability of the server (setup is performed by a single server)	Strongly NP-hard for C_{max}, and the strongly NP-hard for the forced idle time even when setup times are constant, polynomially solvable cases, heuristics
	Kravchenko and Werner (1998)	C_{max} (setup is performed by m-1 servers)	A pseudo-polynomial algorithm
	Schuurman and Woeginger (1999)	C_{max} (preemption)	Algorithm, worst case ratio
	Glass et al. (2000)	C_{max} (setup is performed by a single server, dedicated machines)	NP-hard even for special cases, polynomially solvable cases, algorithms, worst-case ratio
	Hall et al. (2000)	C_{max}, L_{max}, ΣC_j, Σw_jC_j, ΣT_j, Σw_jT_j, ΣU_j, Σw_jU_j (setup is performed by a single server)	Proofs of binary or strongly NP-completeness, polynomial or pseudo-polynomial-time algorithms
	Xing and Zhang (2000)	C_{max} (a job may be processed on two different machines simultaneously)	Heuristic, worst-case performance ratio
	Kravchenko and Werner (2001)	ΣC_j (setup is performed by a single server, unit setup time)	Heuristic, error bound
	Wang and Cheng (2001)	Σw_jC_j (setup is performed by a single server)	An approximation algorithm, worst-case performance
	Abdekhodaee and Wirth (2002)	C_{max} (setup is performed by a single server, only two machines)	Complexity results, integer programming, heuristics
	Brucker et al. (2002)	C_{max}, L_{max}, ΣC_j, Σw_jC_j, ΣT_j, Σw_jT_j, ΣU_j, Σw_jU_j (setup is performed by a single server)	New complexity results for special cases
	Abdekhodaee et al. (2004)	C_{max} (setup is performed by a single server, only two machines, equal setup times)	Complexity results, lower bound, heuristics
	Guirchoun et al. (2005)	C_{max}, L_{max}, ΣC_j, Σw_jC_j, ΣT_j, Σw_jT_j, ΣU_j, Σw_jU_j (setup is performed by a single server)	Complexity results
	Abdekhodaee et al. (2006)	C_{max} (setup is performed by a single server, only two machines)	Greedy heuristic, genetic algorithm
ST_{sd}/ SC_{sd}	Tamimi and Rajan (1997)	Σw_jT_j (Q)	Genetic algorithm
	Heady and Zhu (1998)	Earliness cost + tardiness cost (some machines may not process some jobs)	Heuristic
	Balakrishnan et al. (1999)	$\Sigma w_jE_j + \Sigma w_jT_j$ (Q, r_j)	Mixed integer programming, Bender's decomposition procedure (for larger problems)
	Sivrikaya-Serifoglu and Ulusoy (1999)	$w_E\Sigma E_j + w_T\Sigma T_j$ (two types of uniform parallel machines, r_j)	Genetic algorithm
	Vignier et al. (1999)	Finding a feasible schedule, minimizing sum of costs including the cost of setup times	Heuristic, genetic algorithm, branch-and-bound
	Park et al. (2000)	Σw_jT_j	Neural network, heuristic
	Radhakrishnan and Ventura (2000)	$\Sigma E_j + \Sigma T_j$	Mixed integer programming, simulated annealing
	Zhu and Heady (2000)	$\Sigma w_jE_j + \Sigma w_jT_j$ (R)	Mixed integer programming
	Gendreau et al. (2001)	C_{max}	Lower bounds, heuristic
	Hurink and Knust (2001)	C_{max} (precedence constraints)	Complexity results
	Kurz and Askin (2001)	C_{max} (r_j)	Integer programming, traveling salesman problem, genetic algorithm, multi-fit
	Weng et al. (2001)	Σw_jC_j (R)	Seven heuristics are proposed and evaluated

Hiraishi et al. (2002)	Minimize the weighted number of early and tardy jobs	Establishes that the problem is solvable in polynomial time for certain cases
Mendes et al. (2002b)	C_{max}	Heuristics, tabu search
Fowler et al. (2003)	C_{max}, $\Sigma w_j T_j$, $\Sigma w_j C_j$ (r_j)	Genetic algorithm
Kim et al. (2003b)	$\Sigma w_j T_j$	Heuristic, tabu search
Kim and Shin (2003)	L_{max} (r_j, both identical and non-identical machine cases)	Tabu search
Bilge et al. (2004)	ΣT_j (Q, r_j)	Tabu search
Anglani et al. (2005)	Minimizing the total setup cost (uncertain processing times)	Mixed integer linear programming
Feng and Lau (2005)	$\Sigma w_j E_j + \Sigma w_j T_j$	Squeaky Wheel Optimization heuristic
Nessah et al. (2005)	ΣC_j (r_j)	Sufficient and necessary condition, heuristic, lower bound
Tahar et al. (2006)	C_{max} (job splitting)	Heuristic

Table 4. Parallel machines batch setup time scheduling problems

Setup type	References	Criterion (Comments)	Approach/Result
$ST_{si,b}$/ $SC_{si,b}$	Liaee and Emmons (1997)	ΣC_j (group technology)	NP-hard
	Liu et al. (1999)	ΣC_j ($p_j=p_i$,common setup time)	NP-hard, pseudopolynomial algorithm
	Gambosi and Nicosia (2000)	C_{max} (online scheduling)	Algorithm, upper bound, lower bound
	Webster and Azizoglu (2001)	$\Sigma w_j f_j$	Dynamic programming
	Yi and Wang (2001a)	Σf_j	Heuristic, tabu search
	Yi and Wang (2001b)	Σf_j	Heuristic, lower bound
	Blazewicz and Kovalyov (2002)	ΣC_j ($p_j=p_i$,common setup time, group technology)	Strongly NP-hard, polynomially solvable for a constant number of machines
	Azizoglu and Webster (2003)	$\Sigma w_j f_j$	Branch-and-bound algorithm
	Chen and Powell (2003)	$\Sigma w_j C_j$, $\Sigma w_j U_j$	Branch-and-bound algorithm
	Yi and Wang (2003)	$\Sigma w_j E_j + \Sigma w_j T_j$	A fuzzy logic embedded genetic algorithm (soft computing)
	Wilson et al. (2004)	C_{max} (r_j, common setup time)	Heuristic, genetic algorithm
	Yi et al. (2004)	Σf_j	A fuzzy logic embedded genetic algorithm (soft computing)
	Chen and Wu (2006)	ΣT_j (R, jobs restricted to be processed on certain machines)	Heuristic
	Dunstall and Wirth (2005a)	$\Sigma w_j C_j$	Branch-and-bound, dominance rules
	Dunstall and Wirth (2005b)	$\Sigma w_j C_j$	Heuristics
	Crauwels et al. (2006)	Several performance criteria including to reduce the amount of setups (r_j)	Heuristics, integer programming
	Leung et al. (2006)	ΣC_j (p_j is a step function of the waiting time of job i, common setup time)	Strongly NP-hard, polynomially solvable for equal basic processing times
$ST_{si,b}$, single family, batch availability	Cheng and Kovalyov (2000)	Deadlines (R)	$O(n^{2m+1}/\varepsilon^m)$ approximation scheme, $O(m^2 n^{2m+1})$ for uniform machines, $O(nlogn)$ for identical machines and job processing times equal to the setup time
	Lin and Jeng (2004)	L_{max}, ΣU_i	Pseudopolynomial dynamic programming algorithms for constant number of machines, heuristics
	Yang (2004 a)	ΣC_j (standard and specific components, batching for standard components)	Constructive heuristics
$ST_{sd,b}$/$SC_{sd,b}$	Akkiraju et al. (2001)	$\Sigma w_j T_j$, $\Sigma w_j E_j$, TST (R, additional constraints, multiple objectives)	A heuristic approach called Asynchronous Team architecture to construct Pareto set
	Jeong et al. (2001)	ΣF_j and deviation from product demand (R, additional constraints)	Constructive heuristics
	Eom et al. (2002)	$\Sigma w_j T_j$	Heuristic, tabu search
	Kim et al. (2002)	ΣT_j (R, jobs in a family have the same due date)	Simulated annealing
	Chen and Powell (2003)	$\Sigma w_j C_j$, $\Sigma w_j U_j$	Branch-and-bound algorithm
	Kim et al. (2003a)	$\Sigma w_j T_j$ (R, multioperational jobs, operations of the same job can be processed concurrently)	Constructive heuristics and simulated annealing
	Yalaoui and Chu (2003)	C_{max} (a job can be split into several parts to be processed concurrently)	Heuristic based on a reduction to traveling salesman problem
	Dupuy et al. (2005)	$\Sigma w_j T_j$ (r_j, calendar constraint)	Simulated annealing, new neighborhood mechanisms

Table 5. Flow shop non-batch setup time scheduling problems

Setup type	References	# of stages (type)	Criterion (Comments)	Approach/Result
ST_{si}/SC_{si}	Cheng et al. (1999)	2	C_{max} (R_{si}, setup is performed by a single server)	NP-completeness in the strong sense, heuristics, worst-case error bounds
	Allahverdi (2000)	2	Σf_j	Dominance relations, branch-and-bound, insertion based heuristics
	Allahverdi and Aldowaisan (2000)	3 (no-wait)	ΣC_j	Optimal solutions for certain cases, dominance relation, heuristics
	Botta-Genoulaz (2000)	*m* (flexible)	L_{max} (R_{si}, time lags, precedence constraints)	Heuristics
	Glass et al. (2000)	2	C_{max} (setup is performed by a single server)	NP-hard in the strong sense
	Glass et al. (2000)	2 (no-wait)	C_{max} (setup is performed by a single server)	Polynomial time algorithm
	Sidney et al. (2000)	2 (no-wait)	C_{max} (some setup times consists of two parts)	A heuristic algorithm and its worst-case performance ratio
	Su and Chou (2000)	2	weighted sum of C_{max} and Σf_j	Integer programming, heuristic
	Al-Anzi and Allahverdi (2001)	2	ΣC_j	Heuristics
	Allahverdi and Savsar (2001)	2	C_{max} (machine breakdowns)	Optimal solution for special cases
	Allahverdi and Al-Anzi (2002)	2	L_{max}	Dominance relations, heuristics
	Allahverdi and Aldowaisan (2002)	2	ΣC_j (R_{si})	Optimal solutions for special cases, dominance relations, lower bound, heuristics
	Allahverdi et al. (2003)	2	C_{max}, ΣC_j (random setup times)	Dominance relations
	Aldowaisan and Allahverdi (2004)	3 (no-wait)	ΣC_i (R_{si})	Dominance relation, lower bound, heuristics
	Allaoui and Artiba (2004)	*m* (flexible)	C_{max}, T_{max}, ΣT_j, ΣU_j, ΣC_j (transportation time, breakdowns)	Simulation, optimization, heuristics, simulated annealing
	Brown et al. (2004)	*m* (no-wait)	C_{max}, Σf_j	Non-polynomial time solution methods, heuristic
	Chang et al. (2004b)	2 (no-wait)	Σf_j (R_{si})	Dominance relations, heuristic
	Chang et al. (2004c)	2 (Flexible and no-wait)	C_{max} (R_{si}, one machine at the first stage)	Dominance relations, heuristic
	Dileepan (2004)	2 (no-wait)	L_{max}	Dominance relations
	Shyu et al. (2004)	2 (no-wait)	ΣC_j	Ant colony
	Al-Anzi and Allahverdi (2005a)	2	L_{max}	A novel approach for discovering dominance relations for scheduling problems
	Al-Anzi and Allahverdi (2005b)	2 (assembly)	L_{max}	Dominance relation, self-adaptive differential evolution heuristic
	Allahverdi (2005a)	2	C_{max} (random setup times)	Dominance relations
	Allahverdi (2005b)	2	ΣC_j (random setup times)	Dominance relations
	Allahverdi (2005c)	2	L_{max} (random setup times)	Dominance relations
	Allahverdi et al. (2005)	2	L_{max}	Hybrid genetic algorithm
	Brucker et al. (2005)	*m*	C_{max} , ΣC_j, Σw_jC_j, ΣT_j, Σw_jT_j, Σw_jT_j, L_{max} (setup is performed by a single server)	Complexity results for special cases, NP-hard
	Fondrevelle et al. (2005a)	2 (no-wait)	L_{max} (R_{si})	Branch-and-bound, optimal solutions for certain cases
	Fondrevelle et al. (2005b)	*m*	L_{max}	Dominance relation, lower and upper bounds, branch-and-bound
	Low (2005)	*m* (flexible)	Σf_j (unrelated machines at each stage, R_{sd})	Simulated annealing
	Al-Anzi and Allahverdi (2006)	*m*	Minimizing completion time variance	Hybrid genetic algorithm
	Allahverdi and Al-Anzi (2006a)	3	ΣC_j	Dominance relation, lower bound, branch-and-bound, simulated annealing
	Allahverdi and Al-Anzi (2006b)	2 (assembly)	C_{max}	Dominance relation, heuristics, particle swarm optimization, simulated annealing
	Ng et al. (2006)	3	L_{max}	Dominance relation, heuristics

	Ruiz and Allahverdi (2006)	*m* (no-wait)	ΣC_j	Dominance relation, an iterated local search method
ST_{sd}/ SC_{sd}	Hwang and Sun (1997)	2	C_{max} (precedence constraints, setup times depend on the job before the previous job)	Dynamic programming
	Rajendran and Ziegler (1997)	*m*	$\Sigma w_j f_j$	Heuristic
	Hwang and Sun (1998)	2	C_{max} (precedence constraints, setup times depend on the job before the previous job)	Genetic algorithm
	Sonmez and Baykasoglu (1998)	*m*	$\Sigma w_j T_j$	Dynamic programming
	Bianco et al. (1999)	*m* (no-wait)	C_{max} (r_j)	Lower bounds, mathematical formulation, heuristics
	Norman (1999)	*m*	C_{max} (finite buffer)	Tabu search
	Rios-Mercado and Bard (1999a)	*m*	C_{max}	Branch-and-bound
	Rios-Mercado and Bard (1999b)	*m*	C_{max}	A traveling salesman problem based heuristic
	Demirkol and Uzsoy (2000)	*m* (reentrant)	L_{max}	A series of decomposition methods, tabu search
	Liu and Chang (2000)	*m* (flexible)	Minimizing total setup times and costs (r_j)	Lagrangian relaxation-based integer programming
	Allahverdi and Aldowaisan (2001)	2 (no-wait)	ΣC_j	Optimal solutions for certain cases, dominance relation, heuristics
	Sun and Hwang (2001)	2	C_{max}(setup times only on the second machine, n-step setup times)	Dynamic programming, genetic algorithm
	Tseng and Stafford (2001)	*m*	C_{max}	Mixed-integer linear programming
	Stafford and Tseng (2002)	*m*	C_{max}	Mixed-integer linear programming
	Stafford and Tseng (2002)	*m* (no-wait)	C_{max}	Mixed-integer linear programming
	Kurz and Askin (2003)	*m* (flexible)	C_{max} (some jobs skipping some operations)	Three types of heuristics are explored
	Maddux III and Gupta (2003)	2	C_{max} (some jobs need not to go thorough the second machine)	Lower bound, heuristic
	Rajendran and Ziegler (2003)	*m*	$\Sigma w_j f_j + \Sigma w_j T_j$	Heuristic
	Rios-Mercado and Bard (2003)	*m*	C_{max}	Mixed-integer programming, branch-and-cut algorithm
	Kurz and Askin (2004)	*m* (flexible)	C_{max}	Lower bound, integer programming, heuristics, genetic algorithm
	Pugazhendhi et al. (2004)	*m* (flexible)	$\Sigma w_j f_j$ (non-permutation schedules)	Heuristic
	Andrés et al. (2005b)	*m*	C_{max} and ΣT_j (precedence constraints)	Multi-objective genetic algorithm
	Jungwattanaki et al. (2005)	*m* (flexible)	Minimizing the weighted sum of C_{max} and ΣU_j	Heuristic, genetic algorithm, simulated annealing
	Ruiz et al. (2005a)	*m*	C_{max}	Genetic algorithm
	Ruiz et al. (2005b)	*m* (flexible)	C_{max} (precedence constraints, unrelated machines)	Genetic algorithm
	Ruiz and Maroto (2006)	*m* (flexible)	C_{max}	Genetic algorithm
	Tseng et al. (2005)	*m*	C_{max}	A penalty-based heuristic algorithm
	França et al. (2006)	*m* (no-wait)	C_{max} (r_j)	Hybrid genetic algorithm
	Ruiz and Stützle (2006)	*m*	C_{max} and $\Sigma w_j T_j$	Iterated Greedy algorithms
	Zandieh et al. (2006)	*m* (flexible)	C_{max}	Immune algorithm

Table 6. Flow shop batch setup time scheduling problems

Setup type	References	# of stages (type)	Criterion (Comments)	Approach/Result
$ST_{si,b}$ / $SC_{si,b}$	Cheng and Kovalyov (1998)	3	C_{max} (2 dedicated machines at stage 2)	Dynamic programming
	Huang and Li (1998)	2 (flexible)	C_{max} (one machine at stage 1, group technology)	Heuristics, sequencing rules
	Danneberg et al. (1999)	*m*	C_{max}, $\Sigma w_i C_i$ (batch availability, permutation)	Constructive heuristics and iterative algorithms
	Yang and Chern (2000)	2	C_{max} ($R_{si,b}$, transportation time)	An optimal polynomial time algorithm
	Hall et al. (2003)	*m*	C_{max} (batching within a family, batch availability, no-wait, group technology)	A reduction to a generalized TSP, a customized heuristic
	Reddy and Narendran (2003)	5	Average job time in the system, average tardiness, percentage of tardy jobs (dynamically arriving jobs, stochastic)	Comparison of heuristics for simulated data
	Cheng et al. (2004)	*m+1*	C_{max} (*m* dedicated machines at stage 2)	$O(n^m)$, $O(nlogL)$ for $m=2$
	Pranzo (2004)	2	C_{max} (intermediate buffer of limited capacity, group technology)	Reduction of a special case to a polynomially solvable TSP
	Logendran et al. (2005a)	*m* (flexible)	C_{max} (group technology, several machines in the same stage can process jobs of the same family)	Heuristics
	Wang and Cheng (2005)	2	Σf_j	Properties of the optimal solution, heuristics, branch-and-bound
	Wang and Cheng (2006)	2 (no-wait)	L_{max}	Dominance relations, heuristic
	Yang et al. (2006)	2 (reentrant)	C_{max} (group technology assumption)	NP-hard, branch-and-bound algorithm
$ST_{sd,b}$/ $SC_{sd,b}$	Schaller et al. (2000)	*m*	C_{max} (group technology assumption)	Lower bounds, heuristics
	Cho and Ahn (2003)	2	ΣT_j (group technology)	A hybrid genetic algorithm
	Lin and Liao (2003)	2 (flexible)	T_{max} (setup times on the first stage, one machine at stage 1)	Heuristic
	Andrés et al. (2005a)	3 (flexible)	Minimize setup times	Forming groups, heuristic
	França et al. (2005)	*m*	C_{max} (group technology)	Genetic algorithm, memetic algorithm, multi-start procedure
	Logendran et al. (2005b)	*m* (flexible)	C_{max} (group technology)	Tabu search-based heuristics
	Gupta and Schaller (2006)	*m*	Σf_j (group technology)	Branch-and-bound, heuristics
	Logendran et al. (2006)	2	C_{max} (group technology)	Lower bounds, search algorithms based on tabu search
$ST_{si,b}$, single family, batch availability	Cheng et al. (2000)	2	C_{max} (permutation)	Strongly NP-hard, polynomial algorithms for special cases
	Glass et al. (2001)	2	C_{max}	Strongly NP-hard, 4/3-approximation algorithm
	Lin and Cheng (2001)	2 (no-wait)	C_{max}	Strongly NP-hard, polynomial algorithms for special cases
	Bukchin et al. (2002)	2	ΣC_j (non-integer batch sizes)	Convex programming, optimal solution for a special case
	Kovalyov et al. (2004)	*m+1* (assembly)	C_{max}	A heuristic with tight worst case performance bound of $2\text{-}1/(m+1)$
	Mosheiov et al. (2004)	*m*	C_{max}, ΣC_j , $p_j=p$	$O(n)$, rounding optimal non-integer batch sizes
	Lin and Cheng (2005)	2	C_{max} (no batching on machine 1)	Strongly NP-hard, constructive heuristics, $O(n^2)$ for a special case

Table 7. Job shop scheduling problems

Setup type	References	Criterion (Comments)	Approach/Result
ST_{sd}/SC_{sd}	Sun and Noble (1999)	$\sum w_j T_j^2$ (r_j)	Mathematical formulation
	Focacci et al. (2000)	C_{max}	Branch-and-bound algorithm
	Artigues and Roubellat (2001)	L_{max} (r_j)	Petri net approach
	Cheung and Zhou (2001)	C_{max}	A hybrid genetic algorithm
	Artigues and Roubellat (2002)	L_{max}	Insertion algorithm
	Ballicu et al. (2002)	C_{max}	Mixed integer programming
	Choi and Choi (2002)	C_{max}	Mixed integer programming, a local search algorithm
	Artigues and Buscaylet (2003)	C_{max}	Tabu search
	Sun and Yee (2003)	C_{max} (reentrant job shop)	Heuristics, genetic algorithm
	Artigues et al. (2004)	C_{max}	Branch-and-bound algorithm
	Artigues et al. (2005a)	C_{max}	Upper bounds, priority rule-based multi-pass heuristic
	Artigues et al. (2005b)	C_{max}	Integration of tabu search and multi-pass heuristics
	Balas et al. (2005)	C_{max} (r_j)	Shifting bottleneck procedure, dynamic programming
	Tahar et al. (2005)	$\sum T_j$ (r_j, precedence constraints, hybrid job shop)	Ant colony
	Zoghby et al. (2005)	General (reentrant job shop)	Metaheuristic
$ST_{si,b}$ / $SC_{si,b}$	Sotskov et al. (1999)	Regular and non-regular criteria (r_j)	Insertion algorithm, beam search
	Aldakhilallah and Ramesh (2001)	Batch flowtime and length of one cycle (reentrant job shop)	Constructive heuristics
	Low et al. (2004)	$\boldsymbol{C_{max}}$	Disjunctive graph presentation, integer programming algorithm

Table 8. Open shop scheduling problems

Setup type	References	Criterion (Comments)	Approach/Result
$ST_{sd,b}$	Averbakh et al. (2005)	C_{max} (2-machine, 2 families)	Optimal C_{max} belongs to *[C,(6/5)C]*, where *C* is trivially calculated, *O(n)* to construct a schedule with the makespan from the interval
ST_{si} / SC_{si}	Glass et al. (2000)	C_{max} (setup is performed by a single server)	NP-hard in the strong sense
$ST_{si,b}$ / $SC_{si,b}$	Strusevich (2000)	C_{max} (2-machine)	Worst-case performance ratio, linear-time heuristic
	Glass et al. (2001)	C_{max} (2-machine)	NP-hard in the ordinary sense, heuristic
	Blazewicz and Kovalyov (2002)	$\sum C_j$ (2-machine, group technology)	NP-hard, problems with and without group technology assumption are not equivalent
	Baki and Vickson (2003)	L_{max} (2-machine, continuous presence of an operator)	*O(nlogn)*
	Mosheiov and Oron (2006b)	$C_{max}, \sum C_j$ ($p_j=p$, batch availability)	Constant time for C_{max}, *O(n)* heuristic for $\sum C_i$

References

1. Abdekhodaee, A.H., Wirth, A., 2002. Scheduling parallel machines with a single server: Some solvable cases and heuristics. Computers and Operations Research 29, 295-315.
2. Abdekhodaee, A.H., Wirth, A., Gan, H.S., 2004. Equal processing and equal setup time cases of scheduling parallel machines with a single server. Computers & Operations Research 31, 1867-1889.
3. Abdekhodaee, A.H., Wirth, A., Gan, H.S., 2006. Scheduling two parallel machines with a single server: the general case. Computers & Operations Research 33, 994-1009.
4. Agnetis, A., Alfieri, A., Nicosia, G., 2004. A heuristic approach to batching and scheduling a single machine to minimize setup costs. Computers & Industrial Engineering 46, 793-802.
5. Akkiraju, R. , Keskinocak, P., Murthy, S., Wu, F., 2001. An agent-based approach for scheduling multiple machines. Applied Intelligence 14, 135-144.
6. Al-Anzi, F., Allahverdi, A., 2001. The relationship between three-tiered client-server internet database connectivity and two-machine flowshop. International Journal of Parallel and Distributed Systems and Networks 4, 94-101.
7. Al-Anzi, F., Allahverdi, A., 2005a. Using a hybrid evolutionary algorithm to minimize variance in response time for multimedia object requests. Journal of Mathematical Modelling and Algorithms 4, 435-453.
8. Al-Anzi, F., Allahverdi, A., 2005b. A self-adaptive differential evolution heuristic for two-stage assembly scheduling problem to minimize maximum lateness with setup times. European Journal of Operational Research (to appear).
9. Al-Anzi, F., Allahverdi, A., 2006. Empirically discovering dominance relations for scheduling problems using an evolutionary algorithm. International Journal of Production Research (to appear).
10. Aldakhilallah, K.A., Ramesh, R., 2001. Cyclic scheduling heuristics for a re-entrant job shop manufacturing environment. International Journal of Production Research 39, 2635-2657.
11. Aldowaisan, T., Allahverdi, A., 2004. Three-machine no-wait flowshop with separate setup and removal times to minimize total completion time. International Journal of Industrial Engineering 11, 113-123.
12. Allahverdi, A., 2000. Minimizing mean flowtime in a two-machine flowshop with sequence-independent setup times. Computers & Operations Research 27, 111-127.
13. Allahverdi, A., 2005a. Two-machine flowshop scheduling problem to minimize makespan with bounded setup and processing times. International Journal of Agile Manufacturing 8, 145-153.
14. Allahverdi, A., 2005b. Two-machine flowshop scheduling problem to minimize total completion time with bounded setup and processing times. International Journal of Production Economics (to appear).

15. Allahverdi, A., 2005c. Two-machine flowshop scheduling problem to minimize maximum lateness with bounded setup and processing times. Kuwait Journal of Science and Engineering (to appear).
16. Allahverdi, A. and Al-Anzi, F., 2002. Using two-machine flowshop with maximum lateness objective to model multimedia data objects scheduling problem for WWW applications. Computers and Operations Research 29, 971-994.
17. Allahverdi, A., Al-Anzi, F., 2006a. A branch-and-bound algorithm for three-machine flowshop scheduling problem to minimize total completion time with separate setup times. European Journal of Operational Research 169, 767-780.
18. Allahverdi, A., Al-Anzi, F., 2006b. Evolutionary heuristics and an algorithm for the two-stage assembly scheduling problem to minimize makespan with setup times. International Journal of Production Research (to appear).
19. Allahverdi, A., Al-Anzi, F., Gupta, J.N.D., 2005. A polynomial genetic based algorithm to minimize maximum lateness in a two-stage flowshop with setup times. International Journal of Operations Research 2, 89-100.
20. Allahverdi, A., Aldowaisan, T., 2000. No-wait and separate setup three-machine flowshop with total completion time criterion. International Transactions in Operational Research 7, 245-264.
21. Allahverdi, A., Aldowaisan, T., 2001. Minimizing total completion time in a no-wait flowshop with sequence-dependent additive changeover times. Journal of the Operational Research Society 52, 449-462.
22. Allahverdi, A., Aldowaisan, T., 2002. Two-machine flowshop to minimize total completion time with separated setup and removal times. International Journal of Industrial Engineering 9, 275-286.
23. Allahverdi, A., Aldowaisan, T., Sotskov, Y.N., 2003. Two-machine flowshop scheduling problem to minimize makespan or total completion time with random and bounded setup times. International Journal of Mathematics and Mathematical Sciences 39, 2475-2486.
24. Allahverdi, A., Gupta, J.N.D., Aldowaisan, T., 1999. A review of scheduling research involving setup considerations. OMEGA The International Journal of Management Sciences 27, 219-239.
25. Allahverdi, A., Savsar, M., 2001. Stochastic proportionate flowshop scheduling with setups. Computers & Industrial Engineering 39, 357-369.
26. Allaoui, H., Artiba, A., 2004. Integrating simulation and optimization to schedule a hybrid flow shop with maintenance constraints. Computers & Industrial Engineering 47, 431-450
27. Andrés, C., Albarracín, J.M., Tormo, G., Vicens, E., García-Sabater, J.P., 2005a. Group technology in a hybrid flowshop environment: A case study. European Journal of Operational Research 167, 272-281.
28. Andrés, C., Tomás, J.V., García-Sabater, J.P., Miralles, C., 2005b. A multi-objective genetic algorithm to solve the scheduling problem in flowshops with sequence dependent setup times. Proceedings of the International Conference on Industrial Engineering and Systems

Management, pages 630-635, Marrakech, Morocco, May 16-19, 2005.

29. Anglani, A., Grieco, A., Guerriero, E., Musmanno, R., 2005. Robust scheduling of parallel machines with sequence-dependent set-up costs. European Journal of Operational Research 161, 704-720.
30. Armentano, V.A., Mazzini, R., 2000. A genetic algorithm for scheduling on a single machine with set-up times and due dates. Production Planning and Control 11, 713-720.
31. Artigues, C., Roubellat, F., 2001. A Petri net model and a general method for on and off-line multi-resource shop floor scheduling with setup times. International Journal of Production Economics 74, 63-75.
32. Artigues, C., Roubellat, F., 2002. An efficient algorithm for operation insertion in a multi-resource job-shop scheduling with sequence-dependent setup times. Production Planning and Control 13, 175-186.
33. Artigues, C., Buscaylet, F., 2003. A fast tabu search method for the job-shop problem with sequence-dependent setup times. Proceedings of Metaheuristic International Conference, pages 1-6, Kyoto, Japan, August 25-28, 2003.
34. Artigues, C., Belmokhtar, S., Feillet, D., 2004. A new exact solution algorithm for the job shop problem with sequence-dependent setup times. In J.C. Regin and M. Rueher, editors, 1st International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, Lecture Note in Computer Science 3011, pages 37-49, Springer.
35. Artigues, C., Lopez, P., Ayache, P., 2005a. Scheduling generation schemes for the job-shop problem with sequence-dependent setup times: dominance properties and computational analysis. Annals of Operations Research (to appear).
36. Artigues, C., Buscaylet, F., Feillet, D., 2005b. Lower and upper bounds for the job-shop scheduling problem with sequence-dependent setup times. Proceedings of The 2nd Multidisciplinary International Conference on Scheduling: Theory and Applications, pages 316-321, New York, USA, July 18-21, 2005.
37. Asano, M., Ohta, H., 1999. Scheduling with shutdowns and sequence dependent set-up times. International Journal of Production Research 37, 1661-1676.
38. Aubry, A., Rossi, A., Espinouse, M.L., Jacomino, M., 2006. Minimizing setup costs for parallel multi-purpose machines under load-balancing constraint. European Journal of Operational Research (to appear).
39. Averbakh, I., Berman, O., Chernykh, I., 2005. A 6/5 approximation algorithm for the two-machine routing open-shop problem on a two-node network. European Journal of Operational Research 166, 3-24.
40. Azizoglu, M., Webster, S., 1997. Scheduling job families about an unrestricted common due date on a single machine. International Journal of Production Research 35, 1321-1330.
41. Azizoglu, M., Webster, S., 2003. Scheduling parallel machines to minimize weighted flowtime with family set-up times. International Journal of Production Research 41, 1199-1215.

42. Baker, K.R., 1999. Heuristic procedures for scheduling job families with setups and due dates. Naval Research Logistics 46, 978-991.
43. Baker, K.R., Magazine, M.J., 2000. Minimizing maximum lateness with job families. European Journal of Operational Research 127, 126-139.
44. Baki, M.F., Vickson, R.G., 2003. One-operator, two-machine open shop and flow shop scheduling with setup times for machines and maximum lateness objective. INFOR 41, 301-319.
45. Baki, M.F., Vickson, R.G., 2004. One-operator, two-machine open shop and flow shop problems with setup times for machines and weighted number of tardy jobs objective. Optimization Methods and Software 19, 165-178.
46. Balakrishnan, N., Kanet, J.J., Sridharan, S.V., 1999. Early/tardy scheduling with sequence dependent setups on uniform parallel machines. Computers & Operations Research 26, 127-141.
47. Balas, E., Simonetti, N., Vazacopoulos, A., 2005. Job shop scheduling with setup times, deadlines and precedence constraints. Proceedings of The 2nd Multidisciplinary International Conference on Scheduling: Theory and Applications, pages 520-532, New York, USA, July 18-21, 2005.
48. Ballicu, M., Giua, A., Seatzu, C., 2002. Job-shop scheduling models with set-up times. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics 5, 95-100.
49. Baptiste, P., 2000. Batching identical jobs. Mathematical Methods of Operations Research 52, 355-367.
50. Baptiste, P., Jouglet, A., 2001. On minimizing total tardiness in a serial batching problem. RAIRO Recherche Operationnelle 35, 107-115.
51. Baptiste, P., Le Pape, C., 2005. Scheduling a single machine to minimize a regular objective function under setup constraints. Discrete Optimization 2, 83-99.
52. Berning, G., Brandenburg, M., Gursoy, K., Mehta, V., Tolle, F.J., 2002. An integrated system solution for supply chain optimization in the chemical process industry. OR Spectrum 24, 371-401.
53. Bianco, L., Dell'Olmo, P., Giordani, S., 1999. Flow shop no-wait scheduling with sequence dependent setup times and release dates. INFOR 37, 3-19.
54. Bilge, U., Kirac, F., Kurtulan, M., Pekgun, P. A., 2004. Tabu search algorithm for parallel machine total tardiness problem. Computers & Operations Research 31, 397-414.
55. Blazewicz, J., Kovalyov, M.Y., 2002. The complexity of two group scheduling problems. Journal of Scheduling 5, 477-485.
56. Botta-Genoulaz, V., 2000. Hybrid flow shop scheduling with precedence constraints and time lags to minimize maximum lateness. International Journal of Production Economics 64, 101-111.
57. Brown, S.I, McGarvey, R.G., Ventura, J.A., 2004. Flowtime and makespan for a no-wait m-machine flowshop with set-up times separated. Journal of the Operational Research

Society 55, 614-621.
58. Brucker, P., Dhaenens-Flipo, C., Knust, S., Kravchenko, S.A., Werner, F., 2002. Complexity results for parallel machine problems with a single server. Journal of Scheduling 5, 429-457.
59. Brucker, P., Kovalyov, M.Y., Shafransky, Y.M., Werner, F., 1998. Parallel machine batch scheduling with deadlines and sequence-independent setup times. Annals of Operations Research 83, 23-40.
60. Brucker, P., Knust, S., Wang, G., 2005. Complexity results for flow-shop problems with a single server. European Journal of Operational Research 165, 398-407.
61. Bukchin, J., Tzur, M., Jaffe, M., 2002. Lot splitting to minimize average flow-time in a two-machine flowshop. IIE Transactions 34, 953-970.
62. Bukchin, J., Masin, M., 2004. Multi-objective lot splitting for a single product *m*-machine flow shop line. IIE Transactions 36, 191-202.
63. Chang, P.C., Hsieh, J.C., Wang, Y.W., 2003. Genetic algorithms applied in BOPP film scheduling problems: minimizing total absolute deviation and setup times. Applied Soft Computing 3, 139-148.
64. Chang, T.Y., Chou, F.D., Lee, C.E., 2004a. A heuristic algorithm to minimize total weighted tardiness on a single machine with release dates and sequence-dependent setup times. Journal of the Chinese Institute of Industrial Engineers 21, 289-300.
65. Chang, J., Xing, Z., Shao, H., Pei, B., 2004b. Scheduling a no-wait flowshop with separated setup and removal times. Proceedings of the World Congress on Intelligent Control and Automation (WCICA) 4, 2877-2880.
66. Chang, J., Yan, W., Shao, H., 2004c. Scheduling a two-stage no-wait hybrid flowshop with separated setup and removal times. Proceedings of the American Control Conference 2, 1412-1416, Boston, MA, United States, Jun 30-Jul 2, 2004.
67. Chen, D., Li, S., Tang, G.., 1997. Single machine scheduling with common due date assignment in a group technology environment. Mathematical and Computer Modelling 25, 81-90.
68. Chen, Z.L., Powell, W.B., 2003. Exact algorithms for scheduling multiple families of jobs on parallel machines. Naval Research Logistics 50, 823-840.
69. Chen, J.F., Wu, T.H., 2006. Total tardiness minimization on unrelated parallel machine scheduling with auxiliary equipment constraints. Omega 34, 81-89.
70. Cheng, T.C.E., Chen, Z.L., Oguz, C., 1994. One-machine batching and sequencing of multiple-type items. Computers and Operations Research 21, 717-721.
71. Cheng, T.C.E., Chen, Z.-L., Kovalyov, M.Y., Lin, B.M.T. 1996. Parallel-machine batching and scheduling to minimize total completion time. IIE Transactions 28, 953-956.
72. Cheng, T.C.E., Gupta, J.N.D., Wang, G., 2000. A review of flowshop scheduling research with setup times. Production and Operations Management 9, 262-282.
73. Cheng, T.C.E, Janiak, A., Kovalyov, M.Y., 2001b. Single machine batch scheduling with resource dependent setup and processing times. European Journal of Operational Research 135, 177-183.

74. Cheng, T.C.E., Kovalyov, M.Y., 1998. An exact algorithm for batching and scheduling two part types in a mixed shop: Algorithms and Complexity. International Journal of Production Economics 55, 53-56.
75. Cheng, T.C.E., Kovalyov, M.Y., 2000. Parallel machine batching and scheduling with deadlines. Journal of Scheduling 3, 109-123.
76. Cheng, T.C.E., Kovalyov, M.Y., 2001. Single machine batch scheduling with sequential job processing. IIE Transactions 33, 413-420.
77. Cheng, T.C.E., Kovalyov, M.Y., 2003. Scheduling a single server in a two-machine flow shop. Computing 70, 167-180.
78. Cheng, T.C.E., Kovalyov, M.Y., Chakhlevich, K.N., 2004. Batching in a two-stage flowshop with dedicated machines in the second stage. IIE Transactions 36, 87-93.
79. Cheng, T.C.E, Lin, B.M.T., Toker, A., 2000. Makespan minimization in the two-machine flowshop batch scheduling problem. Naval Research Logistics 47, 128-144.
80. Cheng, T.C.E., Liu, Z., Shafransky, Y.M., 2001a. A note on the complexity of family scheduling to minimize the number of late jobs. Journal of Scheduling 4, 225-229.
81. Cheng, T.C.E., Ng, C.T., Yuan, J.J., 2003a. The single machine batching problem with family setup times to minimize maximum lateness is strongly NP-hard. Journal of Scheduling 6, 483-490.
82. Cheng, T.C.E., Ng, C.T., Yuan, J.J., 2003b. A stronger complexity result for the single machine multi-operation jobs scheduling problem to minimize the number of tardy jobs, Journal of Scheduling 6, 551-555.
83. Cheng, T.C.E., Wang, G., Sriskandarajah, C., 1999. One-operator–two-machine flowshop scheduling with setup and dismounting times. Computers & Operations Research 26, 715-730.
84. Cheung, W., Zhou, H., 2001. Using genetic algorithms and heuristics for job shop scheduling with sequence-dependent setup times. Annals of Operations Research 107, 65-81.
85. Chiu, H.N., Chang, J.H., 2005. Cost models for lot streaming in a multistage flow shop. Omega 33, 435-450.
86. Cho, K.K., Ahn, B.H., 2003. A hybrid genetic algorithm for group scheduling with sequence dependent group setup time. International Journal of Industrial Engineering: Theory Applications and Practice 10, 442-448.
87. Choi, I.C., Choi, D.S., 2002. A local search algorithm for jobshop scheduling problems with alternative operations and sequence-dependent setups. Computers & Industrial Engineering 42, 43-58.
88. Crama, Y., van de Klundert, J., Spieksma, F.C.R., 2002. Production planning problems in printed circuit board assembly. Discrete Applied Mathematics 123, 339-361.
89. Crauwels, H.A.J., Beullens, P., Van Oudheusden, D., 2006. Integrated scheduling and lotsizing with family set-up times on parallel machines. European Journal of Operational Research (to appear).
90. Dang, C., Kang, L., 2004. Batch-processing scheduling with setup times. Journal of

Combinatorial Optimization 8, 137-146.
91. Danneberg, D., Tautenhahn, T., Werner, F., 1999. A comparison of heuristic algorithms for flow shop scheduling problems with setup times and limited batch size. Mathematical and Computer Modelling 29, 101-126.
92. Demirkol, E., Uzsoy, R., 2000. Decomposition methods for reentrant flow shops with sequence-dependent setup times. Journal of Scheduling 3, 155-177.
93. Dileepan, P., 2004. A note on minimizing maximum lateness in a two-machine no-wait flowshop. Computers and Operations Research 31, 2111-2115.
94. Drexl, A., Kimms, A., 1997. Lot sizing and scheduling — Survey and extensions. European Journal of Operational Research 99, 221-235.
95. Dunstall, S., Wirth, A., 2005a. A comparison of branch-and-bound algorithms for a family scheduling problem with identical parallel machines. European Journal of Operational Research 167, 283-296.
96. Dunstall, S., Wirth, A., 2005b. Heuristic methods for the identical parallel machine flowtime problem with set-up times. Computers & Operations Research 32, 2479-2491.
97. Dunstall, S., Wirth, A., Baker, K., 2000. Lower bounds and algorithms for flowtime minimization on a single machine with set-up times. Journal of Scheduling 3, 51-69.
98. Dupuy, M., Lamothe, J., Gaborit, P., Dupont, L., 2005. Efficient neighbors in simulated annealing algorithm to optimize the lead time in a parallel multipurpose machine scheduling problem with setup and calendar constraints. Proceedings of the International Conference on Industrial Engineering and Systems Management, pages 22-31, Marrakech, Morocco, May 16-19, 2005.
99. Ellis, K.P., Lu, Y., Bish, E.K., 2004. Scheduling of wafer test processes in semiconductor manufacturing. International Journal of Production Research 42, 215-242.
100. Eom, D.H., Shin, H.J., Kwun, I.H., Shim, J.K., Kim, S.S., 2002. Scheduling jobs on parallel machines with sequence-dependent family set-up times. International Journal of Advanced Manufacturing Technology 19, 926-932.
101. Eren, T., Guner, E., 2006. A bicriteria scheduling with sequence-dependent setup times. Applied Mathematics and Computation. In Press.
102. Feng, G.., Lau, H.C., 2005. Efficient algorithms for machine scheduling problems with earliness and tardiness penalties. Proceedings of The 2nd Multidisciplinary International Conference on Scheduling: Theory and Applications, pages 196-211, New York, USA, July 18-21, 2005.
103. Focacci, F., Laborie, P., Nuijten, W., 2000. Solving scheduling problems with setup times and alternative resources. Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling, pages 92-101, Breckenbridge, Colorado, USA.
104. Fondrevelle, J., Allahverdi, A., Oulamara, A., 2005a. Two-machine no-wait flowshop scheduling problem to minimize maximum lateness with separate setup and removal times. International Journal of Agile Manufacturing 8, 165-174.
105. Fondrevelle, J., Allahverdi, A., Oulamara, A., Portmann, M.C., 2005b. Permutation

flowshops with exact time lags to minimize maximum lateness. INRIA Intern Research Report, INRIA-Lorraine, Nancy, France.
106. Fowler, J.W., Horng, S.M., Cochran, J.K., 2003. A hybridized genetic algorithm to solve parallel machine scheduling problems with sequence dependent setups. International Journal of Industrial Engineering : Theory Applications and Practice 10, 232-243.
107. França, P.M., Tin Jr, G., Buriol, L.S., 2006. Genetic algorithms for the no-wait flowshop sequencing problem with time restrictions. International Journal of Production Research 44, 939-957.
108. França, P.M., Gupta, J.N.D, Mendes, A., Moscato, P., Veltink, K.J., 2005. Evolutionary algorithms for scheduling a flowshop manufacturing cell with sequence dependent family setups. Computers and Industrial Engineering 48, 491-506.
109. França, P.M., Mendes, A., Moscato, P., 2001. A memetic algorithm for the total tardiness single machine scheduling problem. European Journal of Operational Research 132, 224-242.
110. Gagne, C., Price, W.L., Gravel, M., 2002. Comparing an ACO algorithm with other heuristics for the single machine scheduling problem with sequence-dependent setup times. Journal of the Operational Research Society 53, 895-906.
111. Gambosi, G., Nicosia, G., 2000. On-line scheduling with setup costs. Information Processing Letters 73, 61-68.
112. Garey, M.R., Johnson, D.S., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Co., San Francisco, CA.
113. Gendreau, M., Laporte, G., Guimarães, E.M., 2001. A divide and merge heuristic for the multiprocessor scheduling problem with sequence dependent setup times. European Journal of Operational Research 133, 183-189.
114. Gerodimos, A.E., Glass, C.A., Potts C.N., 2000. Scheduling the production of two-component jobs on a single machine. European Journal of Operational Research 120, 250-259.
115. Gerodimos, A.E., Glass, C.A., Potts C.N., 2001. Scheduling of customized jobs on a single machine under item availability. IIE Transactions 33, 975-984.
116. Gerodimos, A.E., Glass, C.A., Potts C.N., Tautenhahn, T., 1999. Scheduling multi-operation jobs on a single machine. Annals of Operations Research 92, 87-105.
117. Glass, C.A., Shafransky, Y.M., Strusevich, V.A., 2000. Scheduling for parallel dedicated machines with a single server. Naval Research Logistics 47, 304-328.
118. Glass, C.A., Potts, C.N., Strusevich, V.A., 2001. scheduling batches with sequential job processing for two-machine flow and open shops. INFORMS Journal on Computing 13, 120-137.
119. Graham, R.L., Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., 1979. Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of Discrete Mathematics 5, 287-326.
120. Graves, G.H., Lee, C.Y., 1999. Scheduling maintenance and semiresumable jobs on a single

machine. Naval Research Logistics 46, 845-863.
121. Guirchoun, S., Souhkal, A., Martineau, P., 2005. Complexity results for parallel machine scheduling problems with a server in computer systems. Proceedings of The 2nd Multidisciplinary International Conference on Scheduling: Theory and Applications, pages 232-236, New York, USA, July 18-21, 2005.
122. Gupta, J.N.D., Schaller, J.E., 2006. Minimizing flow time in a flow-line manufacturing cell with family setup times. Journal of the Operational Research Society 57, 163-176.
123. Gupta, A.K., Sivakumar, A.I., 2005. Multi-objective scheduling of two-job families on a single machine. Omega 33, 399-405.
124. Gupta, S.R., Smith, J.S., 2006. Algorithms for single machine total tardiness scheduling with sequence dependent setups. European Journal of Operational Research. In Press.
125. Hall, N.G., Laporte, G., Selvarajah, E., Sriskandarajah, C., 2003. Scheduling and lot streaming in flowshops with no-wait in process, Journal of Scheduling 6, 339-354.
126. Hall, N.G., Potts, C.N., Sriskandarajah, C., 2000. Parallel machine scheduling with a common server. Discrete Applied Mathematics 102, 223-243.
127. Havill, J.T., Mao, W., 2006. competitive online scheduling of perfectly malleable jobs with setup times. European Journal of Operational Research (to appear).
128. Heady, R.B., Zhu, Z., 1998. Minimizing the sum of job earliness and tardiness in a multimachine system. International Journal of Production Research 36, 1619-1632.
129. Hiraishi, K., Levner, E., Vlach, M., 2002. Scheduling of parallel identical machines to maximize the weighted number of just-in-time jobs. Computers and Operations Research 29, 841-848.
130. Hochbaum, D.S., Landy, D., 1997. Scheduling with batching: Two job types. Discrete Applied Mathematics 72, 99-114.
131. Huang, W., Li, S., 1998. A two-stage hybrid flowshop with uniform machines and setup times. Mathematical and Computer Modelling 27, 27-45.
132. Hurink, J., Knust, S., 2001. List scheduling in a parallel machine environment with precedence constraints and setup times. Operations Research Letters 29, 231-239.
133. Hwang, H., Sun, J.U., 1997. Production sequencing problem with reentrant work flows and sequence dependent setup times. Computers & Industrial Engineering 33, 773-776.
134. Hwang, H., Sun, J.U., 1998. Production sequencing problem with re-entrant work flows and sequence dependent setup times. International Journal of Production Research 36, 2435-2450.
135. Iravani, S.M.R., Teo, C.P. 2005. Asymptotically optimal schedules for single-server flow shop problem with setup costs and times. Operations Research Letters 33, 421-430.
136. Janiak, A., Kovalyov, M.Y., Portmann, M.C., 2005. Single machine group scheduling with resource dependent setup and processing times. European Journal of Operational Research 162, 112-121.
137. Jeong, B., Kim, S.W., Lee, Y.J., 2001. An assembly scheduler for TFT LCD manufacturing. Computers & Industrial Engineering 41, 37-58.

138. Jungwattanaki, J., Reodecha, M., Chaovalitwongse, P., Werner, F., 2005. An evaluation of sequencing heuristics for flexible flowshop scheduling problems with unrelated parallel machines and dual criteria. Otto-von-Guericke-Universitat Magdeburg, Preprint 28/05, 1-23.
139. Kalir, A.A., Sarin, S.C., 2003. Constructing near optimal schedules for the flow-shop lot streaming problem with sublot-attached setups. Journal of Combinatorial Optimization 7, 23-44.
140. Karabati, S., Akkan, C., 2006. Minimizing sum of completion times on a single machine with sequence-dependent family setup times. Journal of the Operational Research Society 57, 271-280.
141. Karimi, B., Fatemi Ghomi S.M.T., Wilson, J.M., 2003. The capacitated lot sizing problem: a review of models and algorithms. Omega 31, 365-378.
142. Kim, S.C., Bobrowski, P.M., 1997. Scheduling jobs with uncertain setup times and sequence dependency. Omega 25, 437-447.
143. Kim, D.W., Kim, K.H., Jang, W., Chen, F.F., 2002. Unrelated parallel machine scheduling with setup times using simulated annealing. Robotics and Computer-Integrated Manufacturing 18, 223-231.
144. Kim, D.W., Na, D.G., Chen, F.F., 2003a. Unrelated parallel machine scheduling with setup times and a total weighted tardiness objective. Robotics and Computer-Integrated Manufacturing 19, 173-181.
145. Kim, S.S., Shin, H.J., 2003. Scheduling jobs on parallel machines: A restricted tabu search approach. International Journal of Advanced Manufacturing Technology 22, 278-287.
146. Kim, S.S., Shin, H.J., Eom, D.H., Kim, C.O., 2003b. A due date density-based categorizing heuristic for parallel machines scheduling. International Journal of Advanced Manufacturing Technology 22, 753-760.
147. Kleinau, U., 1993. Two-machine shop scheduling problems with batch processing. Mathematical and Computer Modelling 17, 55-66.
148. Kolahan, F., Liang, M., 1998. An adaptive TS approach to JIT sequencing with variable processing times and sequence-dependent setups. European Journal of Operational Research 109, 142-159.
149. Koulamas, C.P., 1996. Scheduling two parallel semiautomatic machines to minimize machine interference. Computers and Operations Research 23, 945-956.
150. Koulamas, C., Kyparisis, G.J., 2006. Single-machine scheduling problems with past-sequence-dependent setup times. European Journal of Operational Research (to appear).
151. Kovalyov, M.Y., Potts, C.N., Strusevich, V.A., 2004. Batching decisions for assembly production systems. European Journal of Operational Research 157, 620-642.
152. Kravchenko, S.A., Werner, F., 1997. Parallel machine scheduling problems with a single server. Mathematical and Computer Modelling 26, 1-11.
153. Kravchenko, S.A., Werner, F., 1998. Scheduling on parallel machines with a single and multiple servers. Otto-von-Guericke-Universitat Magdeburg, Preprint 30/98, 1-18.
154. Kravchenko, S.A., Werner, F., 2001. A heuristic algorithm for minimizing mean flow time

with unit setups. Information Processing Letters 79, 291-296.
155. Kuik, R., Tielemans, P.F.J., 1997. Setup utilization as a performance indicator in production planning and control. International Journal of Production Economics 49, 175-182.
156. Kurz, M.E., Askin, R.G., 2001. Heuristic scheduling of parallel machines with sequence-dependent set-up times. International Journal of Production Research 39, 3747-3769.
157. Kurz, M.E., Askin, R.G., 2003. Comparing scheduling rules for flexible flow lines. International Journal of Production Economics 85, 371-388.
158. Kurz, M.E., Askin, R.G., 2004. Scheduling flexible flow lines with sequence-dependent setup times. European Journal of Operational Research 159, 66-82.
159. Laguna, M., 1999. A heuristic for production scheduling and inventory control in the presence of sequence-dependent setup times. IIE Transactions 31, 125-134.
160. Laporte, G.., 1992. The vehicle routing problem: an overview of exact and approximate algorithms. European Journal of Operational Research 59, 345-358.
161. Lee, S.M., Asllani, A.A., 2004. Job scheduling with dual criteria and sequence-dependent setups: mathematical versus genetic programming. Omega 32, 145-153.
162. Lee, Y. H., Bhaskaran, K., and Pinedo, M., 1997. A heuristic to minimize the total weighted tardiness with sequence dependent setups. IIE Transactions 29, 45-52.
163. Leon, V.J., Peters, B.A., 1998. A comparison of setup strategies for printed circuit board assembly. Computers & Industrial Engineering 34, 219-234.
164. Leu, B.Y., 1999. Comparative analysis of order-input sequencing heuristics in a cellular flexible assembly system for large products. International Journal of Production Research 37, 2861-2873.
165. Leu, B.Y., Wang, F.K., 2000. Analysis of sequencing heuristics in a flexible flow system with hybrid order shipment environments. International Journal of Industrial Engineering: Theory Applications and Practice 7, 67-75.
166. Leung, J.Y-T., Ng, C.T., Cheng, T.C.E., 2006. Minimizing Sum of Completion Times for Batch Scheduling of Jobs with Deteriorating Processing Times. European Journal of Operational Research (to appear).
167. Liaee, M.M., Emmons, H., 1997. Scheduling families of jobs with setup times. International Journal of Production Economics 51, 165-176.
168. Liao, L.M., Liao, C.J., 2002. A tabu search approach for single machine scheduling with major and minor setups. International Journal of Industrial Engineering: Theory Applications and Practice 9, 174-183.
169. Lin, B.M.T., 2002. Fabrication scheduling on a single machine with due date constraints. European Journal of Operational Research 136, 95-105.
170. Lin, B.M.T., Cheng, T.C.E., 2001. Batch scheduling in the no-wait two-machine flowshop to minimize the makespan. Computers & Operations Research 28, 613-624.
171. Lin, B.M.T., Cheng, T.C.E., 2002. Fabrication and assembly scheduling in a two-machine flowshop. IIE Transactions 34, 1015-1020.

172. Lin, B.M.T., Cheng, T.C.E., 2005. Two-machine flowshop batching and scheduling. Annals of Operations Research 133, 149-161.
173. Lin, B.M.T, Jeng, A.A.K., 2004. Parallel-machine batch scheduling to minimize the maximum lateness and the number of tardy jobs. International Journal of Production Economics 91, 121-134.
174. Lin, H.T., Liao, C.J., 2003. A case study in a two-stage hybrid flow shop with setup time and dedicated machines. International Journal of Production Economics 86, 133-143.
175. Liu, C.Y., Chang, S.C., 2000. Scheduling flexible flow shops with sequence-dependent setup effects. IEEE Transactions on Robotics and Automation 16, 408-419.
176. Liu, Z., Cheng, T.C.E., 2002. Scheduling with job release dates, delivery times and preemption penalties. Information Processing Letters 82, 107-111.
177. Liu, Z., Cheng, T.C.E., 2004. Minimizing total completion time subject to job release dates and preemption penalties. Journal of Scheduling 7, 313-327.
178. Liu, Z., Yu, W., 1999. Minimizing the number of late jobs under the group technology assumption. Journal of Combinatorial Optimization 3, 5-15.
179. Liu, Z., Yu, W., Cheng, T.C.E., 1999. Scheduling groups of unit length jobs on two identical parallel machines. Information Processing Letters 69, 275-281.
180. Logendran, R., Carson, S., Hanson, E., 2005a. Group scheduling in flexible flow shops. International Journal of Production Economics 96, 143-155.
181. Logendran, R., deSzoeke, P., Barnard, F., 2005b. Sequence-dependent group scheduling problems in flexible flow shops. International Journal of Production Economics. In Press.
182. Logendran, R., Salmasi, N., Sriskandarajah, C., 2006. Two-machine group scheduling problems in discrete parts manufacturing with sequence-dependent setups. Computers & Operations Research 33, 158-180.
183. Low, C., 2005. Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines. Computers & Operations Research 32, 2013-2025.
184. Low, C., Hsu, C.-M., Huang, K.I., 2004. Benefits of lot splitting in job-shop scheduling International Journal of Advanced Manufacturing Technology 24, 773-780.
185. Maddux III, H.S., Gupta, J.N.D., 2003. Scheduling intermediate and finished products in a two-stage flowshop with sequence dependent setup times. Proceedings - Annual Meeting of the Decision Sciences Institute, 1579-1585.
186. Mason, S.J., Fowler, J.W., Carlyle, W.M., 2002. A modified shifting bottleneck heuristic for minimizing total weighted tardiness in complex job shops. Journal of Scheduling 5, 247-262.
187. Mendes, A.S., França, P.M., Moscato, P., 2002a. Fitness landscapes for the total tardiness single machine scheduling problem. Neural Network World 12, 165-180.
188. Mendes, A.S., Muller, F.M., França, P.M., Moscato, P., 2002b. Comparing meta-heuristic approaches for parallel machine scheduling problems. Production Planning and Control 13, 143-154.
189. Mika, M., Waligora, G., Weglarz, J., 2006. Tabu search for multi-mode resource-constrained project scheduling with schedule-dependent setup times. European Journal of Operational

Research (to appear).
190. Miller, D.M., Chen, H.C., Matson, J., Liu, Q, 1999. A hybrid genetic algorithm for the single machine scheduling problem. Journal of Heuristics 5, 437-454.
191. Monkman, S., Morrice, D., Bard, J., 2006. A production scheduling heuristic for an electronic manufacturer with sequence dependent setup costs. European Journal of Operational Research (to appear).
192. Mosheiov, G., Oron, D., 2005. A note on flow-shop and job-shop batch scheduling with identical processing-time jobs. European Journal of Operational Research 161, 285-291.
193. Mosheiov, G., Oron, D., 2006a. A single machine batch scheduling problem with bounded batch size. European Journal of Operational Research (to appear).
194. Mosheiov, G., Oron, D., 2006b. Open-shop batch scheduling with identical jobs. European Journal of Operational Research (to appear).
195. Mosheiov, G., Oron, D., Ritov, Y., 2004. Flow-shop batch scheduling with identical processing-time jobs. Naval Research Logistics 51, 783-799.
196. Mosheiov, G., Oron, D., Ritov, Y., 2005. Minimizing flow-time on a single machine with integer batch sizes. Operations Research Letters 33, 497-501.
197. Nauss, R.M., 2006. Optimal sequencing in the presence of setup times for tow/barge traffic through a river lock. European Journal of Operational Research (to appear).
198. Nessah, F., Yalaoui, F., Chu, C., 2005. New heuristics for identical parallel machine scheduling with sequence dependent setup times and dates. Proceedings of the International Conference on Industrial Engineering and Systems Management, pages 32-41, Marrakech, Morocco, May 16-19, 2005.
199. Ng, C.T., Allahverdi, A., Al-Anzi, F., Cheng, T.C.E., 2005a. The three-machine flowshop scheduling problem to minimize maximum lateness with separate setup times. International Journal of Operational Research (to appear).
200. Ng, C.T., Cheng, T.C.E., Janiak, A., Kovalyov, M.Y., 2005b. Group scheduling with controllable setup and processing times: minimizing total weighted completion time. Annals of Operations Research 133, 163-174.
201. Ng, C.T., Cheng, T.C.E., Kovalyov, M.Y., 2004. Single machine batch scheduling with jointly compressible setup and processing times. European Journal of Operational Research 153, 211-219.
202. Ng, C.T., Cheng, T.C.E., Yuan, J.J., 2002a. A note on the single machine serial batching scheduling problem to minimize maximum lateness with precedence constraints. Operations Research Letters 30, 66-68.
203. Ng, C.T., Cheng, T.C.E. and Yuan, J.J., 2002b. Strong NP-hardness of the single machine multi-operation jobs total completion time scheduling problem. Information Processing Letters 82, 187-191.
204. Ng, C.T., Cheng, T.C.E., Kovalyov, M.Y., 2003a. Batch scheduling with controllable setup and processing times to minimize total completion time. Journal of the Operational Research Society, 54, 499-506.

205. Ng, C.T., Cheng, T.C.E., Yuan, J.J. and Liu, Z.H., 2003b. On the single machine serial batching scheduling problem to minimize total completion time with precedence constraints, release dates and identical processing times. Operations Research Letters 31, 323-326.
206. Norman, B.A., 1999. Scheduling flowshops with finite buffers and sequence-dependent setup times. Computers & Industrial Engineering 36, 163-177.
207. Norman, B.A., Bean, J.C., 1999. A genetic algorithm methodology for complex scheduling problems. Naval Research Logistics 46, 199-211.
208. Pan, J.C.H., Chen, J.S., Cheng, H.L., 2001. A heuristic approach for single-machine scheduling with due dates and class setups. Computers & Operations Research 28, 1111-1130.
209. Pan, J.C.H., Su, C.S., 1997. Single machine scheduling with due dates and class setups. Journal of the Chinese Institute of Engineers, Transactions of the Chinese Institute of Engineers, Series A/Chung-kuo Kung Ch'eng Hsuch K'an 20, 561-572.
210. Pan, J.C.H., Wu, C.C., 1998. Single machine group scheduling to minimize mean flow time subject to due date constraints. Production Planning and Control 9, 366-370.
211. Park, Y., Kim, S., Lee, Y.H., 2000. Scheduling jobs on parallel machines applying neural network and heuristic rules. Computers & Industrial Engineering 38, 189-202.
212. Pearn, W.L., Chung, S.H., Yang, M.H., 2002a. Minimzing the total machine workload for the wafer probing scheduling problem. IIE Transactions 34, 211-220.
213. Pearn, W.L., Chung, S.H., Yang, M.H., 2002b. The wafer probing scheduling problem (WPSP). Journal of the Operational Research Society 53, 864-874.
214. Pearn, W.L., Chung, S.H., Yang, M.H., Chen, A.Y., 2004a.Algorithms for the wafer probing scheduling problem with sequence-dependent set-up time and due date restrictions . Journal of the Operational Research Society 55, 1194-1207.
215. Pearn, W.L., Chung, S.H., Chen, A.Y., Yang, M.H., 2004b. A case study on the multistage IC final testing scheduling problem with reentry. International Journal of Production Economics 88, 257-267.
216. Pinedo, M., 1995. Scheduling, Theory, Algorithms, and Systems, Prentice Hall, New Jersey, USA.
217. Potts, C.N., Kovalyov, M.Y., 2000. Scheduling with batching: A review. European Journal of Operational Research 120, 228-349.
218. Pranzo, M., 2004. Batch scheduling in a two-machine flow shop with limited buffer and sequence independent setup times and removal times. European Journal of Operational Research 153, 581-592.
219. Pugazhendhi, S., Thiagarajan, S., Rajendran, C., Anantharaman, N., 2004. Generating non-permutation schedules in flowline-based manufacturing sytems with sequence-dependent setup times of jobs: A heuristic approach. The International Journal of Advanced Manufacturing Technology 23, 64-78.
220. Rabadi, G., Mollaghasemi, M., Anagnostopoulos, G.C., 2004. A branch-and-bound algorithm for the early/tardy machine scheduling problem with a common due-date and

sequence-dependent setup time. Computers & Operations Research 31, 1727-1751.
221. Radhakrishnan, S., Ventura, J. A., 2000. Simulated annealing for parallel machine scheduling with earliness-tardiness penalties and sequence-dependent set-up times. International Journal of Production Research 38, 2233-2252.
222. Rajendran, C., Ziegler, H., 1997. A heuristic for scheduling to minimize the sum of weighted flowtime of jobs in a flowshop with sequence-dependent setup times of jobs. Computers & Industrial Engineering 33, 281-284.
223. Rajendran, C., Ziegler, H., 2003. Scheduling to minimize the sum of weighted flowtime and weighted tardiness of jobs in a flowshop with sequence-dependent setup times. European Journal of Operational Research 149, 513-522.
224. Reddy, V., Narendran, T.T., 2003. Heuristics for scheduling sequence-dependent set-up jobs in flow line cells. International Journal of Production Research 41, 193-206.
225. Ríos-Mercado, R.Z., Bard, J.F., 1999a. A branch-and-bound algorithm for permutation flow shops with sequence-dependent setup times. IIE Transactions 31, 721-731.
226. Ríos-Mercado, R.Z., Bard, J.F., 1999b. An enhanced TSP-based heuristic for makespan minimization in a flow shop with setup times. Journal of Heuristics 5, 53-70.
227. Ríos-Mercado, R.Z., Bard, J.F., 2003. The flow shop scheduling polyhedron with setup times. Journal of Combinatorial Optimization 7, 291-318.
228. Ruiz, R., Allahverdi, A., 2006. Some effective heuristics for no-wait flowshops with setup times. Annals of Operations Research (to appear).
229. Ruiz, R., Maroto, C., 2006. A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. European Journal of Operational Research 169, 781-800.
230. Ruiz, R., Stützle, T., 2006. An iterated greedy heuristic for the sequence dependent setup times flowshop with makespan and weighted tardiness objectives. European Journal of Operational Research (to appear).
231. Ruiz, R., Maroto, C., Alcaraz, J., 2005a. Solving the flowshop scheduling problem with sequence dependent setup times using advanced metaheuristics. European Journal of Operational Research 165, 34-54.
232. Ruiz, R., Sivrikaya-Serifoglu, F., Urlings, T., 2005b. Evolutionary approach to realistic hybrid flexible flowshop scheduling problems. Technical Report. Polytechnic University of Valencia. Dept. of Applied Statistics and Operations Research, Spain.
233. Schaller, J., 2006. Scheduling on a single machine with family setups to minimize total tardiness. International Journal of Production Economic. In Press.
234. Schaller, J., Gupta, J.N.D., 2006. Single machine scheduling with family setups to minimize total earliness and tardiness. European Journal of Operational Research (to appear).
235. Schaller, J., Gupta, J.N.D., Vakharia, A.J., 2000. Scheduling a flowline manufacturing cell with sequence dependent family setup times. European Journal of Operational Research 125, 324-339.

236. Schultz, S.R., Hodgson, T.J., King, R.E., Taner, M.R., 2004. Minimizing L_{max} for the single machine scheduling problem with family set-ups. International Journal of Production Research 42, 4315-4330.
237. Schuurman, P., Woeginger, G.J., 1999. Preemptive scheduling with job-dependent setup times. Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, 759-767.
238. Shin, H.J., Kim, C.O., Kim, S.S., 2002. A tabu search algorithm for single machine scheduling with release times, due dates, and sequence-dependent set-up times. International Journal of Advanced Manufacturing Technology 19, 859-866.
239. Shufeng, W., Yiren, Z., 2002. Scheduling to minimize the maximum lateness with multiple product classes in batch processing. Proceedings of IEEE TENCON'02, 1595-1598.
240. Shyu, S.J., Lin, B.M.T., Yin, P.Y., 2004. Application of ant colony optimization for no-wait flowshop scheduling problem to minimize the total completion time. Computers & Industrial Engineering 47, 181-193.
241. Sidney, J.B., Potss, C.N., Sriskandarajah, C., 2000. A heuristic for scheduling two-machine no-wait flow shops with anticipatory setups. Operations Research Letters 26, 165-173.
242. Simons, J.V., Russel, G.R., 2002. A case study of batching in a mass service operation. Journal of Operations Management 20, 577-592.
243. Sivrikaya-Serifoglu, F., Ulusoy, G., 1999. Parallel machine scheduling with earliness and tardiness penalties. Computers & Operations Research 26, 773-787.
244. Sonmez, A.I., Baykasoglu, A., 1998. New dynamic programming formulation of (n×m) flowshop sequencing problems with due dates. International Journal of Production Research 36, 2269-2283.
245. Soric, K., 2000a. The CLWS heuristic for single machine scheduling problem. European Journal of Operational Research 120, 352-358.
246. Soric, K., 2000b. A cutting plane algorithm for a single machine scheduling problem. European Journal of Operational Research 127, 383-393.
247. Sotskov, Y.N., Tautenhahn, T., Werner, F., 1999. On the application of insertion techniques for job shop problems with setup times. RAIRO Recherche Operationnelle 33, 209-245.
248. Sourd, F., 2005. Earliness–tardiness scheduling with setup considerations. Computers & Operations Research 32, 1849-1865.
249. Stafford, E.F., Tseng, F.T., 2002. Two models for a family of flowshop sequencing problems. European Journal of Operational Research 142, 282-293.
250. Strusevich, V.A., 2000. Group technology approach to the open shop scheduling problem with batch setup times. Operations Research Letters 26, 181-192.
251. Su, L.H., Chou, F.D., 2000. Heuristic for scheduling in a two-machine bicriteria dynamic flowshop with setup and processing times separated. Production Planning and Control 11, 806-819.
252. Sun, J.U., Hwang, H., 2001. Scheduling problem in a two-machine flow line with the N-step prior-job-dependent set-up times. International Journal of Systems Science 32, 375-385.
253. Sun, X, Noble, J.S., 1999. An approach to job shop scheduling with sequence-dependent

setups. Journal of Manufacturing Systems 18, 416-430.

254. Sun, X., Noble, J.S., Klein, C.M., 1999. Single-machine scheduling with sequence dependent setup to minimize total weighted squared tardiness. IIE Transactions 31, 113-124.
255. Sun, J.U., Yee, S.R., 2003. Job shop scheduling with sequence dependent setup times to minimize makespan. International Journal of Industrial Engineering: Theory Applications and Practice 10, 455-461.
256. Suriyaarachchi, R.H., Wirth, A., 2004. Earliness/tardiness scheduling with a common due date and family setups. Computers & Industrial Engineering 47, 275-288.
257. Tahar, D.N., Yalaoui, F., Chu, C., Amodeo, L., 2006. A linear programming approach for identical parallel machine scheduling with job splitting and sequence-dependent setup times. International Journal of Production Economics 99, 63-73.
258. Tahar, D.N., Yalaoui, F., Amodeo, L., Chu, C, 2005. An ant colony system minimizing total tardiness for hybrid job shop scheduling problem with sequence dependent setup times and release dates. Proceedings of the International Conference on Industrial Engineering and Systems Management, pages 469-478, Marrakech, Morocco, May 16-19, 2005.
259. Tamimi, S.A., Rajan, V.N., 1997. Reduction of total weighted tardiness on uniform machines with sequence dependent setups. Industrial Engineering Research - Conference Proceedings, 181-185.
260. Tan, K.C., Narasimhan, R., 1997. Minimizing tardiness on a single processor with sequence-dependent setup times: a simulated annealing approach. Omega 25, 619-634.
261. Tan, K.C., Narasimhan, R., Rubin, P.A., Ragatz, G.L., 2000. A comparison of four methods for minimizing total tardiness on a single processor with sequence dependent setup times. Omega 28, 313-326.
262. Tseng, F.T., Gupta, J.N.D, Stafford, E.F., 2005. A penalty-based heuristic algorithm for the permutation flowshop scheduling problem with sequence-dependent set-up times. Journal of the Operational Research Society 57, 541-551.
263. Tseng, F.T., Stafford, E.F., 2001. Two MILP models for the NXM SDST flowshop sequencing problem. International Journal of Production Research 39, 1777-1809.
264. Tovey, C.A., 2004. Non-approximability of precedence-constrained sequencing to minimize setups. Discrete Applied Mathematics 134, 351-360.
265. Valls, V., Perez, M.A., Quintanilla, M.S., 1998. A tabu search approach to machine scheduling. European Journal of Operational Research 106, 277-300.
266. Van Der Veen, J.A.A., Woeginger, G.J., Zhang, S., 1998. Sequencing jobs that require common resources on a single machine: A solvable case of the TSP. Mathematical Programming, Series B 82, 235-254.
267. Van Hop, N., Nagarur, N.N., 2004. The scheduling problem of PCBs for multiple non-identical parallel machines. European Journal of Operational Research 158, 577-594.
268. Van Oyen, M.P., Duenyas, I., Tsai, C.Y., 1999. Stochastic sequencing with job families, set-up times, and due dates. International Journal of Systems Science 30, 175-181.
269. Vieira, G. E., Herrmann, J. W., Lin, E., 2000. Analytical models to predict the performance of

a single-machine system under periodic and event-driven rescheduling strategies. International Journal of Production Research 38, 1899-1915.
270. Vignier, A., Sonntag, B., Portmann, M.C., 1999. Hybrid method for a parallel-machine scheduling problem. IEEE Symposium on Emerging Technologies and Factory Automation, ETFA 1, 671-678.
271. Wagelmans, A.P.M, Gerodimos, A.E., 2000. Improved dynamic programs for some batching problems involving maximum lateness criterion. Operations Research Letters 27, 109-118.
272. Wang, G., Cheng, T.C.E., 2001. An approximation algorithm for parallel machine scheduling with a common server. Journal of the Operational Research Society, 52, 234-237.
273. Wang, X., Cheng, T.C.E., 2005. Two-machine flowshop scheduling with job class setups to minimize total flowtime. Computers & Operations Research 32, 2751-2770.
274. Wang, X., Cheng, T.C.E., 2006. A heuristic approach for tow-machine no-wait flowshop scheduling with due dates and class setups. Computers & Operations Research 33, 1326-1344.
275. Wang, L., Wang, M., 1997. Hybrid algorithm for earliness-tardiness scheduling problem with sequence dependent setup time. Proceedings of the IEEE Conference on Decision and Control 2, 1219-1222.
276. Wang, S., Zou, Y., 2002. Scheduling to minimize the maximum lateness with multiple product classes in batch processing. IEEE Region 10 Annual International Conference, Proceedings/TENCON 3, 1595-1598.
277. Webster, S., Azizoglu, M., 2001. Dynamic programming algorithms for scheduling parallel machines with family setup times. Computers & Operations Research 28, 127-137.
278. Webster, S., Jog, P.D., Gupta, A., 1998. Genetic algorithm for scheduling job families on a single machine with arbitrary earliness/tardiness penalties and an unrestricted common due date. International Journal of Production Research 36, 2543-2551.
279. Weng, M.X., Lu, J., Ren, H., 2001. Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective. International Journal of Production Economics 70, 215-226.
280. Wilson, A.D., King, R.E., Hodgson, T.J., 2004. Scheduling non-similar groups on a flow line: Multiple group setups. Robotics and Computer-Integrated Manufacturing 20, 505-515.
281. Woeginger, G.J., 1998. A polynomial-time approximation scheme for single-machine sequencing with delivery times and sequence-independent batch set-up times. Journal of Scheduling 1, 79-87.
282. Xing, W., Zhang, J., 2000. Parallel machine scheduling with splitting jobs. Discrete Applied Mathematics 103, 259-269.
283. Yalaoui, F., Chu, C., 2003. An efficient heuristic approach for parallel machine scheduling with job splitting and sequence-dependent setup times. IIE Transactions 35,183-190.
284. Yang, W.H., 2004a. Scheduling two-component products on parallel machines. Omega 32, 353-359.
285. Yang, W.H., 2004b. Optimal scheduling of two-component products on a single facility.

International Journal of Systems Science 35, 49-53.
286. Yang, D.L., Chern, M.S., 2000. Two-machine flowshop group scheduling problem. Computers & Operations Research 27, 975-985.
287. Yang, D.L., Kuo, W.H., Chern, M.S., 2006. A two-machine multi-family scheduling problem with reentrant production flows. European Journal of Operational Research (to appear).
288. Yang, W.H., Liao, C.J., 1998. Batching and sequencing of jobs with order availability at a single facility. International Journal of Systems Science 29, 13-20.
289. Yang, W.H., Liao, C.J., 1999. Survey of scheduling research involving setup times. International Journal of Systems Science 30, 143-155.
290. Yang, W.H., Chand, S., 2006. Learning and forgetting effects on a group scheduling problem. European Journal of Operational Research (to appear).
291. Yang, M.H., Pearn, W.L., Chung, S.H., 2002. A case study on the wafer probing scheduling problem. Production Planning and control 13, 66-75.
292. Yi, Y., Chang, H.Y., Wang, J., Bai, J.C., 2004. Soft computing for parallel scheduling with setup times. Proceedings of International Conference on Machine Learning and cybernetics, 2041-2046.
293. Yi, Y., Wang, D.W., 2001a. Tabu search for scheduling grouped jobs on parallel machines. Journal of Northeastern University 22, 188-191.
294. Yi, Y., Wang, D.W., 2001b. Scheduling grouped jobs on parallel machines with setups. Computer Integrated Manufacturing Systems 7, 7-11.
295. Yi, Y., Wang, D.W., 2003. Soft computing for scheduling with batch setup times and earliness-tardiness penalties on parallel machines. Journal of Intelligent Manufacturing 14, 311-322.
296. Yokoyama, M., 2006. Flow-shop scheduling with setup and assembly operations. European Journal of Operational Research (to appear).
297. Yuan, J.J., Jang, A.F., Cheng, T.C.E., 2004. A note on the single machine serial batching scheduling problem to minimize maximum lateness with identical processing times. European Journal of Operational Research 158, 525-528.
298. Yuan, J.J., Liu, Z.H., Ng, C.T., Cheng, T.C.E., 2005a. Single machine batch scheduling problem with family setup times and release dates to minimize makespan. Journal of Scheduling (to appear).
299. Yuan, J.J., Lin, Y.X., Cheng, T.C.E., Ng, C.T., 2005b. Single machine serial-batching scheduling problem with common batch size to minimize total weighted completion time. International Journal of Production Economics (to appear).
300. Zandieh, M., Fatemi Ghomi, S.M.T., Moattar Husseini, S.M., 2006. An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times. Applied Mathematics and Computation. In Press.
301. Zhu, Z., Heady, R.B., 2000. Minimizing the sum of earliness/tardiness in multi-machine scheduling: a mixed integer programming approach. Computers & Industrial Engineering 38, 297-305.

302. Zoghby, J., Barnes, J.W., Hasenbein, J.J., 2005. Modeling the reentrant job shop scheduling problem with setups for metaheuristic searches. European Journal of Operational Research 167, 336-348.