Journal of the Korean Institute of Industrial Engineers
Vol. 35, No. 4, pp. 257-265, December 2009.

Surrogate Objective based Search Heuristics to Minimize the Number of Tardy Jobs for Multi-Stage Hybrid Flow Shop Scheduling

Hyun-Seon Choi[1] · Hyung-Won Kim[2] · Dong-Ho Lee[2†]

[1]KMA Consultants, Inc., Youngdeungpo-gu, Seoul, 150-869, Korea
[2]Department of Industrial Engineering, Hanyang University, Seongdong-gu, Seoul, 133-791, Korea

다 단계 혼합흐름공정 일정계획에서 납기지연 작업 수의 최소화를 위한 대체 목적함수 기반 탐색기법

최현선[1] · 김형원[2] · 이동호[2†]

[1]한국능률협회컨설팅 / [2]한양대학교 산업공학과

This paper considers the hybrid flow shop scheduling problem for the objective of minimizing the number of tardy jobs. In hybrid flow shops, each job is processed through multiple production stages in series, each of which has multiple identical parallel machines. The problem is to determine the allocation of jobs to the parallel machines at each stage as well as the sequence of the jobs assigned to each machine. Due to the complexity of the problem, we suggest search heuristics, tabu search and simulated annealing algorithms with a new method to generate neighborhood solutions. In particular, to evaluate and select neighborhood solutions, three surrogate objectives are additionally suggested because not much difference in the number of tardy jobs can be found among the neighborhoods. To test the performances of the surrogate objective based search heuristics, computational experiments were performed on a number of test instances and the results show that the surrogate objective based search heuristics were better than the original ones. Also, they gave the optimal solutions for most small-size test instances.

Keywords: Hybrid Flow Shops, Scheduling, Number of Tardy Jobs, Search Heuristics, Surrogate Objectives

1. Introduction

A hybrid flow, a system with serial production stages and one or more parallel machines at each stage, is an extended system of the ordinary flow shop. The parallel machines are generally added to increase system throughput as well as flexibility. In other words, it is a natural way to increase the system capacity by adding more machines to each production stage of the ordinary flow shop (Gupta 1988). In the hybrid flow shop, the flow of jobs is unidirectional through the serial production stages and each job can be processed by one of the parallel machines at each stage. Various hybrid flow shops can be found in the electronics industry such as printed circuit board (PCB), semiconductor, and lead frame manufacturing (Linn and Zhang 1999; Lee *et al*. 2004). Besides

This work was supported by Brain Korea 21 Grant funded by Korea Government, This is gratefully acknowledged.
† Corresponding author : 133-791 Department of Industrial Engineering, Hanyang University, Seongdong-gu, Seoul, Korea
Fax: +82-2-2297-0475, E-mail: leman@hanyang.ac.kr
Received July 30, 2009; Revision Received October 8, 2009; Accepted November 11, 2009.

these, various traditional industries, such as food, chemical and steel, have various types of hybrid flow shops (Tsubone *et al*. 1996).

There are a number of previous research articles on hybrid flow shop scheduling, which can be classified according to the performance measures used: those without due-dates and those with due-dates. (See Linn and Zhang (1999) for a literature review.) Gupta and Tunc (1991) considered a two-stage hybrid flow shop scheduling problem with parallel machines only at the second stage and suggested heuristics that minimize makespan. Other heuristics for the objective of minimizing makespan in two-stage hybrid flow shops were suggested by Lee and Vairaktarakis (1994), Chen (1995), Lee and Park (1999), and Choi *et al*. (2009). Fouad *et al*. (1998) considered a three-stage hybrid flow shop scheduling problem for the woodworking industry and suggested heuristics that minimize makespan. Brah and Hunsucker (1991) considered the multi-stage hybrid flow shop scheduling problem and suggested a branch and bound algorithm that minimizes makespan, and later, their lower bounds were improved by Moursli and Pochet (2000). For the same multi-stage problem, Guinet and Solomon (1996) suggested the list scheduling algorithms in which the jobs are listed in some order using priority rules and then assigned to the machines according to this order, and Janiak *et al*. (2007) suggested search heuristics under the multiple objectives of minimizing the total weighted earliness, tardiness and waiting time, and Kemal *et al*. (2007) suggested an ant colony algorithm for the multi-stage problem with the makespan measure. Also, Azizoglu *et al*. (2001) considered the objective of minimizing the total flow time for multi-stage hybrid flow shops and suggested a branch and bound algorithm that gives the optimal solutions.

Unlike the articles without due-dates, not much work has been done on the problems with due-date based performance measures due to their problem complexities. Lee and Kim (2004) considered a two-stage hybrid flow shop with parallel machines only at the first stage and suggested a branch and bound algorithm that minimizes the total tardiness. Later, Lee *et al*. (2004) extended their earlier research to multi-stage hybrid flow shops and suggested a bottleneck-focused heuristic in which a schedule for the bottleneck stage is constructed and then those for the other stages are constructed based on that for the bottleneck, and Lee (2006) suggested a list scheduling approach for the problem with dynamic order arrival. Gupta and Tunc (1998) suggested heuristics for a two-stage problem that minimizes the number of tardy jobs. Recently, Choi and Lee (2007) considered a general two-stage hybrid flow shop with two or more parallel identical machines at both stages, and suggested a branch and bound algorithm that minimizes the number of tardy jobs, and later, Choi and Lee (2009) improved their branch and bound algorithm by tightening the lower bounds and suggested heuristics for large-size problems. For the objective of minimizing the maximum tardiness, Guinet and Solomon (1996) suggested the list scheduling algorithms for multi-stage hybrid flow shops.

We consider multi-stage hybrid flow shop scheduling that determines the allocation of jobs to parallel machines at each stage as well as the sequence of the jobs allocated to each machine for the objective of minimizing the number of tardy jobs. The objective considered here is important in many practical cases since the cost penalty incurred by a tardy job does not depend on how late it is, but the event that it is late. For example, a late job may cause a customer to switch to another supplier, especially in the just-in-time production environment (Ho and Chang 1995). In overall, this research is a generalization of Choi and Lee (2007, 2009) in that three or more production stages are considered for hybrid flow shop scheduling. To the best of the authors' knowledge, there are no previous research articles on multi-stage hybrid flow shop scheduling that minimizes the number of tardy jobs.

The problem considered in this paper is NP-hard, which can be easily seen from the fact that the parallel machine scheduling problem that minimizes the number of tardy jobs is NP-hard (Garey and Johnson 1979). In fact, Choi and Lee (2007) reported that their branch and bound algorithms can give the optimal solutions for the test instances only with 14 jobs even in two-stage hybrid flow shops. Therefore, to obtain good quality solutions within a reasonable amount of computation time, we suggest search heuristics, tabu search and simulated annealing algorithms, each of which incorporates a new method to generate neighborhood solutions. In particular, three surrogate objectives are additionally suggested to evaluate and select neighborhood solutions because not much difference in the number of tardy jobs can be found among neighborhood solutions. To the best of our knowledge, there is no previous research on applying surrogate objectives to hybrid flow shop scheduling that minimizes the number of tardy jobs. To show the performances of the surrogate objective based search heuristics, computational experiments were done on a number of test instances and the results are reported.

The rest of this paper is organized as follows. The following section describes the problem with a mathematical formulation. Section 3 presents the search heuristics with the method to generate neighborhood solutions and the surrogate objectives to evaluate and select neighborhood solutions. The test results on computational experiments are reported in

Section 4, and finally, Section 5 concludes the paper with the discussion of future research.

2. Problem description

Before describing the problem in more detail, we explain the structure of a multi-stage hybrid flow shop. As shown in <Figure 1>, the multi-stage hybrid flow shop consists of two or more serial stages and there exist one or more identical parallel machines at each stage. In this figure, m_k denotes the number of machines at stage k, $k = 1, \cdots, K$. Note that we consider the general hybrid flow shop in that there may exist one or more parallel machines at each stage, i.e., $m_k \geq 1$ for $k = 1, \ldots, K$. Each job has K operations and operation k is processed on one of the parallel machines at the stage k.

As stated earlier, the problem considered here has two decisions : (a) allocating jobs to parallel machines at each stage; and (b) sequencing the jobs allocated to each machine. The objective is to minimize the number of tardy jobs, i.e.,

$$\text{minimize} \sum_{i=1}^{n} \delta(T_i),$$

where $T_i = \max\{0, c_{iK} - d_i\}$ and $\delta(T_i) = 1$ if $T_i > 0$, and 0 otherwise. Here, c_{iK} and d_i denote completion time of job i at the last stage K and due-date of job i, respectively. The completion time of each job depends on the two decision variables, allocation and sequencing, and our problem is to determine the two variables that minimize the number of tardy jobs. As stated earlier, this research extends the previous research articles to multi-stage hybrid flow shops with three or more serial production stages.

In this paper, we consider a static and deterministic version of the problem. That is, all jobs are ready for processing at time zero and the job descriptors, such as processing times and due-dates, are deterministic and given in advance. Also, each operation of a job has the same processing time on each parallel machine since we consider identical parallel machines at each stage. Other assumptions made for the problem considered here are: (a) each machine can process only one job at a time and each job can be processed only on a machine at each stage; (b) setup times for the jobs are sequence-independent and hence can be included in the corresponding processing times; (c) no job can be split and pre-emptied; and (d) machine breakdowns are not considered.

3. Solution algorithms

Two types of search heuristics, tabu search and simulated annealing algorithms, are presented in this section. First, the algorithm to obtain an initial solution is explained. Then, the search heuristics, together with the neighborhood generation method and the surrogate objectives, are explained.

3.1 Obtaining an initial solution

The initial solution is obtained using the list scheduling ap-

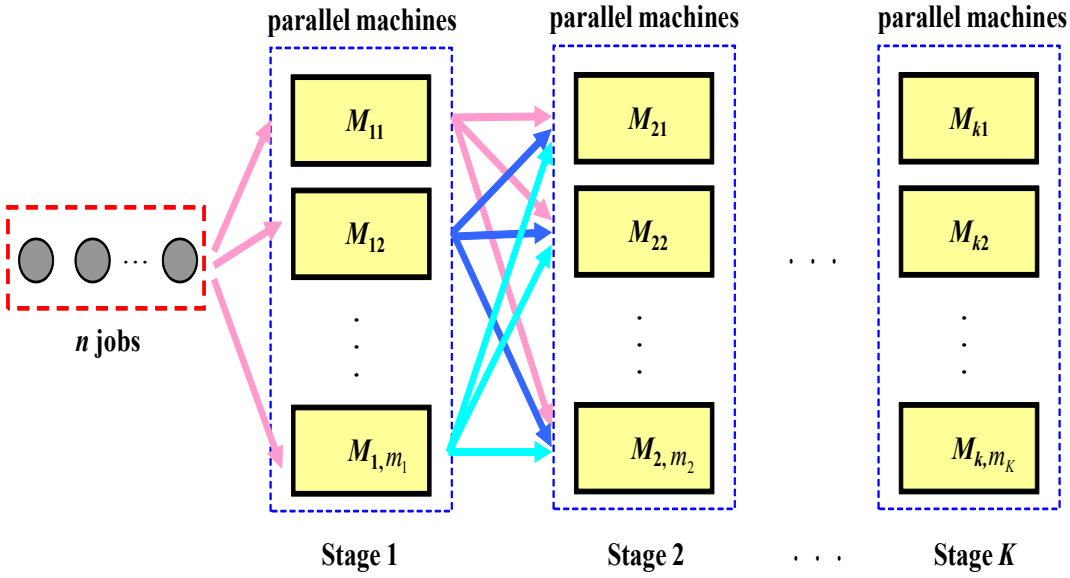

Figure 1. A schematic description of multi-stage hybrid flow shops

proach in which all jobs are listed in some order using a priority rule and then assigned to the machines according to this order. To order the jobs, the EDD (earliest due date) rule is used. Then, according to this order, the jobs are allocated to the earliest available machine from the first to the last stage. Here, the earliest available machine is the one that has the smallest completion time before the job is allocated.

3.2 Search heuristics

3.2.1 Tabu search algorithms

Tabu search (TS) is a well-known search technique to escape from terminating at local optimum prematurely (Glover 1989). In general, TS can be described as a process which attempts to move from the current solution to one of its neighbors. The moves that are worse than the current solution might be accepted to escape from the entrapment of a local optimum in its search for the global optimum. The most recent moves are classified as tabu (forbidden) for a particular number of iterations (tabu tenure or tabu list size) to avoid the cyclic searching path, and these moves are stored in a set Λ, called the *tabu list*. In other words, the elements of Λ define all tabu moves that cannot be applied to the current solution. The size of Λ is bounded by a parameter l, called the *tabu list size*. If $|\Lambda| = l$, before adding a move to Λ, one must remove an element in it, the oldest one in general. Note that a tabu move can be allowed to be chosen if it creates a solution better than the incumbent solution, called the *aspiration criterion* in the literature. See Glover (1989) and Glover and Laguna (1993) for generic descriptions of the TS algorithm.

An application of TS is characterized by *representation of solutions*, *generation of neighbourhood solutions*, *definition of tabu moves*, and *termination condition*. A detailed explanation of each component is explained below.

(a) *Solution representation.* The solution is represented by a set of vectors ($S_1, S_2, \cdots, S_K$), where S_k denotes a permutation of n jobs to be allocated to the parallel machines at stage k. (This solution representation method is also used for the SA algorithms.) Note that we focus on non-permutation schedules in which the job sequences are allowed to be different at different stages. According to the given sequence for each stage, the job allocation is done in such a way that each job is allocated and sequenced to the corresponding earliest available machine. As defined earlier, the earliest available machine implies the one that has the smallest completion time before a job is allocated.

For example, consider a two-stage hybrid flow shop problem with 8 jobs and 2 machines at both stages. Let the solution be S_1 = (1, 2, 3, 4, 5, 6, 7, 8) and S_2 = (1, 2, 3, 4, 5, 7, 8, 6). Then, we can easily see that the resulting schedule, in which the jobs are allocated to the earliest available machine according to the job sequences S_1 and S_2, can be represented as <Figure 2>.

(b) *Neighborhood generation.* Neighborhood solutions are generated using a hybrid interchange and insertion method. Here, the interchange method generates a neighborhood solution by selecting two jobs in the current sequence and interchanging them, while the insertion method selects two jobs in the current sequence and removes the first job from the original place and inserting it to the position that directly precedes the second job. In this research, we use the hybrid method to generate neighborhood solutions as scattered as possible, called the *diversification strategy* in the literature. See Kim *et al.* (2007) for other application of the hybrid neighborhood generation method.

The hybrid method to generate neighborhood solutions can be described as <Figure 3>. Let ($S_1, S_2, \cdots, S_K$) denote the current job sequences at serial stages. As can be seen in the figure, the hybrid method uses the interchange and the insertion methods in a consecutive way. (More specifically, the insertion method is done for a specified number of times after the interchange method is used. In this research, the specified number was set to the tabu list size l from a preliminary test.) The detailed explanation of the hybrid method is as follows.

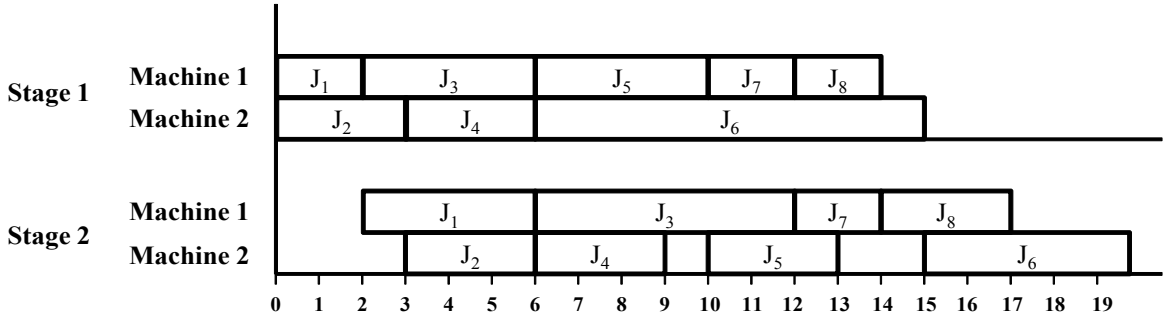

Figure 2. Complete schedule for a given solution representation : an example

First, a new job sequence S_k' for a randomly selected stage k is obtained by interchanging two randomly selected jobs in the current sequence S_k. Then, the insertion method is applied to the new job sequences $(S_1, \cdots, S_k', \cdots, S_k)$ for a specified number of times in such a way that a job with the maximum flow time is selected and it is placed to a randomly selected position located earlier than the current position. More formally, selected is the job i^* at stage k^* such that

$$(i^*,\ k^*) = \underset{i \geq 1,\ k > 1}{\arg\max} \{C_{ik} - r_{ik}\},$$

where C_{ik} and r_{ik} denote the completion time and the ready time of job i at stage k, respectively. Here, the ready time of a job at a stage implies the time at which the job is available for processing at the stage.

There may be a number of ways to consider the moves. Among them, we use the method of examining a portion of the entire neighborhood and taking the best move that is not tabu. This is because the hybrid method described earlier may generate too many neighborhood solutions, and hence it is necessary to limit the number of neighborhood solutions examined. This is called the *candidate list strategy* in the literature (Laguna *et al*. 1991). The purpose of the candidate list strategy is to screen the neighborhood solutions in order to concentrate on promising moves. From a preliminary test, the number of neighborhood solutions examined was set to the tabu list size l. Note that the candidate list is maintained for neighborhood solutions generated by the hybrid method with insertion and interchange operator.

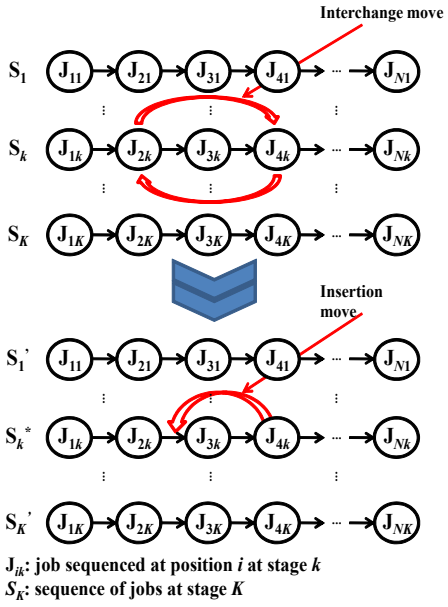

Figure 3. Hybrid neighborhood generation method

(c) *Neighborhood selection with surrogate objectives.* The objective function considered here, minimizing the number of tardy jobs, converges to the local optimum easily, i.e., not much difference in the objective values of neighborhood solutions. To overcome this difficulty, in this paper, we suggest surrogate objectives to evaluate and select a neighborhood solution among those that give the same smallest number of tardy jobs. The following are three surrogate objectives tested.

- *Minimum tardiness* : select a neighborhood that has the job with the minimum tardiness
- *Minimum mean tardiness* : select a neighborhood that gives the minimum mean tardiness
- *Maximum mean earliness* : select a neighborhood that gives the maximum mean earliness

In summary, three TS algorithms, i.e., TS1, TS2, and TS3 with the surrogate objectives of minimum tardiness, minimum mean tardiness, and maximum mean earliness, are suggested in this paper.

(d) *Tabu conditions.* Tabu moves are defined as follows. In the interchange method, a tabu move is defined as a pair of jobs that have been interchanged and the corresponding stage. Also, the insertion method defines a tabu move as the job to be moved and the job that directly precedes the second job at the corresponding stage. More specifically, if job i is inserted between jobs $(j-1)$ and j at stage k, jobs i and j together with stage k, i.e., (i, j, k), are stored in the tabu list. If job i is inserted to the last position in the job sequence, job i and the last job together with the corresponding stage are defined as a tabu move. As stated earlier, a tabu move can be allowed to be chosen if it generates a solution better than the incumbent solution, i.e., the best objective value obtained so far.

(e) *Termination condition.* The three TS algorithms stop if no improvements have been made for a certain number of consecutive iterations, denoted by L_{TS} in this paper. Here, the iteration number increases by one whenever an interchange move is done.

3.2.2 Simulated annealing algorithms

Like the TS, the SA is a search heuristic that attempts to move from the current solution to one of its neighbours. Starting from an initial solution, SA generates a new solution in the neighborhoods of the original one. Then, the change in the objective function value is calculated. (In our application, the surrogate objectives were used to evaluate the objective

value since there is not much difference in the objective values of neighborhood solutions.) If there is an improvement, the transition to the new solution is accepted. Otherwise, the transition to the new solution is accepted with a specified probability denoted by $\exp(-\Delta/t)$, where Δ is the amount of change in the surrogate objective value and t is a control parameter called the *temperature*. By allowing the uphill moves (that increase a surrogate objective value), the SA can escape from a local minimum.

There are four generic parameters to implement the SA algorithms: *initial temperature* (t_0); *epoch length* (α), i.e., number of transitions made with the same temperature; *rule specifying how the temperature is reduced*; and *termination condition*. A choice of these parameters is referred to as a *cooling schedule*, and the performance of an SA algorithm is affected by these parameters and methods. In our application, the method suggested by Park and Kim (1998) was used to set the parameters required. Also, the temperature was decreased using the commonly used equation, $t_k = r \cdot t_{k-1}$, where t_k is the temperature used during the kth epoch and r is a positive constant, called the *cooling ratio*, with a value less than 1. Finally, we terminated the algorithm when there is no improvement for a certain number of iterations, denoted by L_{SA} in this paper.

As in the TS algorithms, we tested three SA algorithms according the three surrogate objectives, i.e., SA1, SA2 and SA3 with the surrogate objectives of minimum tardiness, minimum mean tardiness, and maximum mean earliness, together with the hybrid neighborhood generation method explained earlier. In fact, the SA algorithms are identical to the TS algorithms except for the basic search mechanism.

4. Computational experiments

To compare the performances of the search heuristics suggested in this paper, computational tests were done, and the results are reported in this section. We tested eight search heuristics, TS0 (SA0) without the surrogate objectives and TS1 (SA1), TS2 (SA2), TS3 (SA3) with the surrogate objectives of minimum tardiness, minimum mean tardiness, and maximum mean earliness, respectively. All the algorithms were coded in C and the tests were done on a workstation with an Intel Xeon processor operating at 3.20 GHz clock speed and 1.0 GB RAM memory.

To find the appropriate values of the parameters, a preliminary experiment was done for each search heuristic type. For the TS algorithms, several values for the tabu list size (l) and L_{TS} for the termination condition were tested on representative test instances, and they were set to 50 and 500, respectively. Also, for the SA algorithms, the parameters were set as (t_0, α, r, L_{SA}) = (1, 20, 0.99, 500) according to the method suggested by Park and Kim (1998), where t_0, α, r, and L_{SA} denote the initial temperature, the epoch length, the cooling ratio, and the parameter for the termination condition, respectively.

The first test is on the comparison of the search heuristics with each other for multi-stage hybrid flow shops since the optimal solutions could not be obtained in a reasonable amount of computation time. The performance measures used are: (a) the relative performance ratio; and (b) CPU seconds. Here, the relative performance ratio of search heuristic a for a problem is defined as

$$\left[(C_a - C_{best}) - C_{best}\right] \cdot 100(\%)$$

where C_a is the original objective value obtained from algorithm a and C_{best} is the best original objective value for that problem among the four search heuristics.

For the test, 300 instances were generated, i.e., 50 instances for each of six combinations of three levels of the number of jobs (20, 50, and 100) and two levels of the number of stages (5 and 7). The number of parallel machines at each stage and the processing times were generated from DU (1, 5) and DU (10, 40), respectively. Here, DU (a, b) is the discrete uniform distribution with range [a, b]. Finally, the due-dates were generated using a modified one of the method of Gupta (1988) since it considers two-stage hybrid flow shop. More formally, they were generated from DU ($p \cdot \alpha$, $p \cdot \beta$), where α and β ($\beta > \alpha$) were set to 0.2 and 0.4 and p was set as

$$p = \left\{ \sum_{j=1}^{k} \left(\sum_{i=1}^{n} p_{ij}/m_j \right) + (n-1) \cdot \max_j \left[\sum_{i=1}^{n} p_{ij}/m_j \right] \right\} / \max$$

Test results for multi-stage hybrid flow shops are given in <Table 1> that summarizes the average relative performance ratios and CPU seconds. It can be seen from the table that the TS algorithms are better than the SA algorithms in overall solution quality. In addition, the TS algorithms required much shorter computation times than the SA algorithms under the same termination condition. Also, the search heuristics with surrogate objectives outperformed those without surrogate objectives, which shows the effectiveness of the surrogate objectives for hybrid flow shop scheduling that minimizes the number of tardy jobs. Therefore, it is recommended that the surrogate objectives (instead of the original objective to break ties among the neighbourhood solutions with the same number of tardy jobs) be used for other scheduling problems that minimize the number of tardy jobs. Finally, among the three surrogate objectives, minimizing the

Table 1. Test results for multi-stage hybrid flow shops

Number of jobs	Number of stages	TS0[1]	TS1[2]	TS2[3]	TS3[4]	SA0[1]	SA1[2]	SA2[3]	SA3[4]
20	5	0.23 (3)*	0.15 (3)	0.16 (3)	0.11 (3)	0.23 (3)*	0.19 (4)	0.21 (4)	0.17 (4)
	7	0.23 (4)	0.16 (5)	0.21 (5)	0.09 (4)	0.29 (7)	0.21 (6)	0.23 (5)	0.13 (5)
50	5	0.25 (35)	0.10 (37)	0.14 (47)	0.04 (37)	0.31 (35)	0.26 (42)	0.19 (56)	0.09 (39)
	7	0.26 (48)	0.11 (44)	0.13 (49)	0.08 (43)	0.28 (59)	0.22 (48)	0.25 (52)	0.14 (49)
100	5	0.22 (582)	0.12 (532)	0.14 (531)	0.03 (414)	0.27 (942)	0.20 (1010)	0.23 (1220)	0.21 (1329)
	7	0.22 (673)	0.12 (594)	0.16 (664)	0.05 (555)	0.28 (1215)	0.22 (1103)	0.17 (1183)	0.12 (1416)
Average		0.24 (219)	0.13 (202)	0.16 (217)	0.07 (176)	0.28 (377)	0.22 (369)	0.22 (420)	0.14 (474)

[1] search heuristics without surrogate objective
[2] search heuristics with the surrogate objective of minimum tardiness
[3] search heuristics with the surrogate objective of minimum mean tardiness
[4] search heuristics with the surrogate objective of maximum mean earliness
* average relative performance ratio out of 50 problems and CPU seconds (in parenthesis)

maximum mean earliness gave better results than the others. This may be because of the fact that earliness is closely related with slack time and hence the surrogate objective of the maximum mean earliness may decrease the possibility of being tardy. On the other hand, the surrogate objectives associated with tardiness are directly related to the number of tardy jobs and hence break ties randomly.

The second test was done for two-stage hybrid flow shops. In this test, TS3 and SA3 with the surrogate objective of maximum mean earliness were compared with the optimal branch and bound algorithm of Choi and Lee (2009). (Here, we selected TS3 and SA3 since they perform better than the

Table 2. Test results for two-stage hybrid flow shops：

(a) Cases of loose due-dates

(M_1, M_2)[1]	SA3				TS3			
	Number of jobs				Number of jobs			
	10	12	14	15	10	12	14	15
(1, 2)	$10^2/10^3$	10/10	10/10	10/10	10/10	10/10	10/10	10/10
(1, 3)	10/10	10/10	10/10	10/10	10/10	10/10	10/10	10/10
(1, 4)	10/10	10/10	10/10	9/9	10/10	10/10	10/10	9/9
(2, 2)	10/10	10/10	10/10	10/10	10/10	10/10	10/10	9/10
(2, 3)	10/10	10/10	10/10	10/10	10/10	10/10	10/10	10/10
(2, 4)	10/10	10/10	10/10	10/10	10/10	10/10	10/10	10/10
(3, 2)	10/10	10/10	10/10	10/10	10/10	10/10	10/10	10/10
(3, 3)	10/10	10/10	10/10	10/10	10/10	10/10	10/10	10/10
(3, 4)	10/10	10/10	10/10	10/10	10/10	10/10	10/10	10/10
(4, 2)	10/10	10/10	10/10	9/10	10/10	10/10	10/10	10/10
(4, 3)	10/10	10/10	10/10	10/10	10/10	10/10	10/10	10/10
(4, 4)	10/10	10/10	10/10	10/10	10/10	10/10	10/10	10/10

[1] number of machines at the first and the second stages, respectively
[2] number of problems that the search heuristics gave the optimal solutions out of 10 problems
[3] number of problems that the branch and bound algorithm gave the optimal solutions out of 10 problems

(b) Cases of tight due-dates

(M_1, M_2)[1]	SA3				TS3			
	Number of jobs				Number of jobs			
	10	12	14	15	10	12	14	15
(1, 2)	10/10*	10/10	10/10	8/8	10/10	10/10	10/10	8/8
(1, 3)	10/10	10/10	10/10	9/9	10/10	10/10	10/10	9/9
(1, 4)	10/10	10/10	9/9	8/8	10/10	10/10	9/9	8/8
(2, 2)	10/10	10/10	10/10	9/9	10/10	10/10	10/10	9/9
(2, 3)	10/10	10/10	10/10	10/10	10/10	10/10	10/10	10/10
(2, 4)	10/10	10/10	10/10	9/10	10/10	10/10	10/10	10/10
(3, 2)	10/10	10/10	10/10	10/10	10/10	10/10	10/10	10/10
(3, 3)	10/10	10/10	10/10	10/10	10/10	10/10	10/10	10/10
(3, 4)	10/10	10/10	10/10	10/10	10/10	10/10	10/10	10/10
(4, 2)	10/10	10/10	10/10	10/10	10/10	10/10	10/10	10/10
(4, 3)	10/10	10/10	10/10	10/10	10/10	10/10	10/10	10/10
(4, 4)	10/10	10/10	10/10	10/10	10/10	10/10	10/10	10/10

See the foot notes of Table (a).

others.) The performance measures are: (a) number of instances that each search heuristic gave the optimal solutions; and (b) CPU seconds. For this test, we used the 960 test instances of Choi and Lee (2009), i.e., 10 instances for each of 96 combinations of the number of machines (1, 2, 3 and 4 at the first stage and 2, 3 and 4 at the second stage), four levels of the number of jobs (10, 12, 14 and 15), and two levels of the due-date tightness (loose and tight).

Test results are summarized in <Table 2>, which shows that TS3 and SA3 suggested in this paper give the optimal solutions for most test instances. Note that the CPU seconds are not reported here since they solved all the test problems within 2 seconds. On the other hand, the branch and bound algorithm required more than 2000 seconds to obtain the optimal solutions for the test problems with 15 jobs (Choi and Lee 2009). Therefore, we can conclude that TS3 and SA3 can be used instead of the optimal branch and bound algorithm for small-size two-stage hybrid flow shop scheduling problems since they can give very near optimal solutions while requiring much shorter computation times.

5. Concluding remarks

This paper considered multi-stage hybrid flow shop scheduling, which can be found usually in the electronics industry as well as various traditional industries such as food, chemical and steel. The problem is to determine the allocation of jobs to parallel machines at each stage as well as the sequence of the jobs assigned to each machine for the objective of minimizing the number of tardy jobs. Due to the complexity of the problem, two types of search heuristics, tabu search and simulated annealing algorithms, were suggested with a hybrid neighborhood generation method. In particular, surrogate objectives were additionally suggested to break ties among the neighborhood solutions with the same number of tardy jobs since the objective has the characteristic that not much difference can be found among the neighborhood solutions. The computational results on a number of test problems can be summarized as follows. First, the search heuristics with surrogate objectives outperformed those without surrogate objectives. Second, the tabu search algorithm with the surrogate objective of the maximum mean earliness outperformed the others. Finally, for two-stage hybrid flow shops, the best search heuristics with the surrogate objective of minimizing the maximum mean earliness can be used instead of the optimal branch and bound algorithm.

As an initial research on multi-stage hybrid flow shop scheduling that minimizes the number of tardy jobs, this research has certain further research issues. First, it may be needed to develop an optimal solution algorithm, especially in the theoretical aspect. Second, the lower bounds are worth to be developed to evaluate the absolute solution qualities of

possible future heuristics to be developed. Finally, the problem can be extended to those with uniform or unrelated parallel machines at each stage and/or re-entrant product flows for more practical applications.

References

Azizoglu, M., Cakmak, E. and Kondakci, S. A. (2001), A flexible flow shop problem with total flow time minimization, *European Journal of Operational Research*, **132**, 528-538.

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

Chen, B. (1995), Analysis of classes of heuristics for scheduling a two-stage flow shop with parallel machines at on stage, *Journal of Operation Research Society*, **46**, 231-244.

Choi, H.-S. and Lee, D.-H. (2007), A branch and bound algorithm for two-stage hybrid flow shops: minimizing the number of tardy jobs, *Journal of the Korean Institute of Industrial Engineers*, **33**, 213-220.

Choi, H.-S., Kim, H.-W., Lee, D.-H., Yun, J., Yoon, C. Y., and Chae, K. B. (2009), Scheduling algorithms for two-stage reentrant hybrid flow shops: minimizing makespan under the maximum allowable due-dates, *International Journal of Advanced Manufacturing Technology*, **42**, 963-973.

Choi, H.-S. and Lee, D.-H. (2009), Scheduling algorithms to minimize the number of tardy jobs in two-stage hybrid flow shops, *Computers and Industrial Engineering*, **56**, 113-120.

Fouad, R., Abdelhakim, A. and Salah, E. E. (1998), A hybrid three-stage flowshop problem: efficient heuristics to minimize makespan, *European Journal of Operational Research*, **109,** 321-329.

Garey, M. R. and Johnson, D. S. (1979), *Computers and Intractability*: *A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company.

Glover, F. (1989), Tabu search: part I, *ORSA Journal of Computing*, **1**, 190-206.

Glover, F., Laguna, M. (1993), Tabu search in Modern Heuristics Techniques for Combinatorial Problems, *Blackwell Scientific Publications*, Oxford, 70-141.

Guinet, A. G. P. and Solomon, M. M. (1996), Scheduling hybrid flow shops to minimize maximum tardiness or maximum completion time, *International Journal of Production Research*, **34**, 1643-1654.

Gupta, J. N. D. (1988), Two-stage hybrid flow shop scheduling problem, *Journal of Operation Research Society*, **39**, 359-364.

Gupta, J. N. D. and Tunc, E. A. (1991), Scheduling for a two-stage hybrid flowshop with parallel machines at the second stage, *International Journal of Production Research*, **29**, 1480-1502.

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

Ho, J. C. and Chang, Y. L. (1995), Minimizing the number of tardy jobs for m parallel machines, *European Journal of Operational Research*, **84,** 343-355.

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

Kemal, A., Orhan, E. and Alper, D. (2007), Using ant colony optimization to solve hybrid flow shop scheduling problems, *International Journal of Advanced Manufacturing Technology*, **35,** 541-550.

Kim, S.-I., Choi, H,-S. and Lee, D.-H. (2007), Scheduling algorithms for parallel machines with sequence-dependent setup and distinct ready times: minimizing total tardiness, *Proceedings of the Institution of Mechanical Engineers Part B*: *Journal of Engineering Manufacture*, **221**, 1087-1096.

Laguna, M., Barnes, J. W. and Glover, F. (1991), Tabu search methods for a single machine scheduling problem, *Journal of Intelligent Manufacturing*, **2**, 63-74.

Lee, C. Y. and Vairaktarakis, G. L. (1994), Minimizing makespan in hybrid flow shops, *Operations Research Letters*, **16**, 149-158.

Lee, G.-C. (2006), Scheduling methods for a hybrid flowshop with dynamic order arrival, *Journal of the Korean Institute of Industrial Engineers*, **32**, 373-381.

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

Lee, G.-C., Kim, Y.-D. and Choi, S.-W. (2004), Bottleneck-focused scheduling for a hybrid flow shop, *International Journal of Production Research*, **42**, 165-181.

Lee, J. S. and Park, S. H. (1999), Scheduling heuristics for a two-stage hybrid flowshop with nonidentical parallel machines, *Journal of the Korean Institute of Industrial Engineers*, **25,** 254-265.

Linn, R. and Zhang, W. (1999), Hybrid flow shop scheduling, *Computers and Industrial Engineering*, **37**, 57-61.

Mourisli, O. and Pochet, Y. (2000), A branch-and-bound algorithm for the hybrid flow shop, *International Journal of Production Research Economics*, **64**, 113-125.

Park, M.-W. and Kim, Y.-D. (1998), A systematic procedure for setting parameters in simulated annealing algorithms, *Computers and Operations Research*, **25**, 207-217.

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