Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004



# Scheduling a two-stage no-wait hybrid flowshop with separated setup and removal times

Junlin Chang, Weiwu Yan, Huihe Shao

*Abstract*—**The paper studies the two-stage no-wait hybrid flowshop scheduling problem where one of the two stages consists of several identical parallel machines and the setup and removal times of each job at each stage are separated from the processing time. In view of the NP-complete of this problem, heuristic algorithms regarding sequencing and assigning of jobs to each stage as independent procedure are proposed. Two theorems are also proposed for sequencing job procedure. Computational experience demonstrates the effectiveness of the heuristic algorithm including the proposed theorems in finding a near optimal schedule.**

## I. Introduction

We consider the two-stage no-wait hybrid flowshop scheduling problem which occurs when the operations of a job has to be processed from start to end without interruptions on or between machines. A two-stage no-wait hybrid flowshop consists of two stages and stage $j$ has $m_j$ identical parallel machines ( $j = 1,2$ ). $n$ Given jobs $\{J_i \mid 1 \le i \le n\}$ are to be processed on the two stages in the same technological order, first on stage1 and second on stage 2. An operation of any job consists of three phases: setup, processing, and removal. As described in [6], the setup phase immediately precedes the processing phase, and the removal phase immediately follows the processing phase. The setup phase of an operation on any machine can only start after the removal phase of its predecessor on that machine has been completed. However, a job can be moved to its next operation without waiting for its removal operation to be completed at the current stage. The processing phases of a job in the two stages are not allowed to overlap; the other phases may overlap. Since setup times are considered separate from processing times, the setup work of a job on a subsequent machine can be performed while it is idle before the job arrives on the machine. The setup, processing, and removal times of the operation of job $J_i$ on stage $j$ are denoted by $s_{i,j}$ , $t_{i,j}$ , and $r_{i,j}$ respectively.

Manuscript received September 20, 2003. This work was supported in part by the National 863 Project Fund of P. R. China under Grant 2002 AA412010.

J. Chang is with the Department of Automation, Shanghai Jiaotong University, Shanghai, 200030, P. R. China (phone: 0086-21-62933427-13; e-mail: cjl@ sjtu.edu.cn).
W. Yan is with the Department of Automation, Shanghai Jiaotong University, 200030, P. R. China (email: yanwwsjtu@hotmail.com).
H. Shao is with the Faculty of Department of Automation, Shanghai Jiaotong University, 200030, P. R. China (e-mail: hhshao@ sjtu.edu.cn).

Unlike the classical flowshop where unlimited intermediate storage space is available to hold partially completed jobs between the two stages, in no-wait flowshop the processing phase of any job in the second stage must start exactly at the time the processing phase of that job in the first stage is completed. These conditions are quite common in several industries such as metallurgical, plastic, and chemical production process. For instance, in the case of steel production, the heated metal must continuously go through a sequence of operations before it is cooled down so as to prevent defects in the composition of the material [15]. Other applications can also be found in just-in-time and flexible manufacturing systems. The no-wait scheduling problems have attracted the attention of many researchers both in practical application area and in theoretical area.

The no-wait hybrid flowshop is a generalization of the no-wait pure flowshop and the identical parallel machine shop. The no-wait pure flowshop has been studied extensively by many researchers; see Aldowaisan [1], Aldowaisan and Allahverdi [2], [16], [17], Allahverdi, [3], Gupta [6]. The no-wait hybrid flowshop ignored the setup and removal times have been studied by Salvador [10], Sriskandarajah [13] and Liu [15]. Salvador has developed a branch and bound algorithm to find minimum finish time for a no-wait flowshop with parallel machines model that arise in an actual application in the synthetic fiber industry. The worst case and average case analysis of some heuristic algorithms of this problem has been carried out in Sriskandarajah [13], Liu [15].

This paper considers the two-stage no-wait hybrid flowshop scheduling problems where the first stage contains only one machine and the second stage contains more than one identical parallel machine. The objective is to minimize the makespan of all jobs, i.e. the total throughput time in which all jobs complete processing on both stages. For

0-7803-8335-4/04/$17.00 ©2004 AACC

general case, the no-wait hybrid flowshop is NP-complete even ignored the setup and removal times (Sriskandarajah and Ladet [11]), efficient optimal algorithms for minimizing makespan are not likely to exist. Although it is possible to develop an algorithm using branch-and-bound techniques, it is time consuming even for a moderate scale problem. Thus, heuristic algorithms are inevitably required to obtain good solutions in reasonable computing time instead of looking for exact optimization algorithms. Encouraged by earlier work of two-stage no-wait flow shop study and the two-stage hybrid flowshop without no-wait restriction research, we propose two heuristic algorithms to solve the two-stage no-wait hybrid flowshop problems.

## II. NOTATION AND FORMULATION

Assume the number of jobs to be scheduled is $n$ and the number of machines in the second stage is $m$. To compute the makespan of a schedule, following Gupta [6], consider a permutation $S$ of $n$ jobs. Let $A_1(i)$ be the time job $J_i$ completes processing at stage 1 and $B_{k,1}(i)$ be the time job $J_i$ completes its processing at stage 2, if $J_i$ is processed on machine $k$ at stage 2. Let the immediately preceding job on the machine $k$ at stage 2 be $J_l$, preceding job at stage 1 be $J_j$. Let $A_2(i)$ and $B_{k,2}(i)$ be the specific machine availability times after processing job $J_i$ at stages 1 and 2, respectively. We introduce a dummy job $J_0$ with $A_1(0)=A_2(0)=B_1(0)=B_2(0)=0$. Then the problem can be formulated as,

$$A_1(i)=\max\{A_2(j)+s_{i,1}+t_{i,1}, B_{k,2}(l)+s_{i,2}\} \quad (1)$$

$$A_2(i)=A_1(i)+r_{i,1}$$

$$B_{k,1}(i)=A_1(i)+t_{i,2}$$

$$B_{k,2}(i)=B_{k,1}(i)+r_{i,2}$$

For a schedule $S$, assume that the last job assigned on machine $k$ in stage 2 is job $l_k$. The machine-based makespan $M(S)$ is defined as

$$M(S)=\max_{1\le k\le m}\{B_{k,2}(l_k)\} \quad (2)$$

The solution of the two-stage no-wait hybrid flowshop scheduling problem requires two aspects: sequencing jobs on both stages and assignment of jobs to various machines at each stage. One of approximate approaches is to assume the sequencing and assignment of jobs to machines at each stage can be done independently. This kind of heuristic is two-phase algorithm. First, they order the job using a sequencing no-wait pure flowshop algorithm with two machines. In order to take into account the machines in second stage, $s_{i,2}$, $t_{i,2}$, and $r_{i,2}$ should be divided by the machines number $m$. Second, assigned the jobs to the machines: a job is assigned to first stage and the last free machine in second stage in the sorted order to minimize its final completion time satisfying the no-wait restriction.

## III. SEQUENCING THE TWO-STAGE NO-WAIT PURE FLOWSHOP

We assume that every job requires processing on both machines, so it is necessarily that such a two-stage no-wait flowshop is a permutation flowshop. Let $s_{[i],k}$, $t_{[i],k}$, and $r_{[i],k}$ be the setup, processing and removal times of the job in position $i$ of the sorted order on machine $k$ ($k=1 or 2$). Defining $C_{[i]}$ to be the completion time of the job in position $i$, thus the makespan $M(S)=C_{[n]}$.

$$C_{[1]}=\max\{s_{[1],2}, s_{[1],1}+t_{[1],1}\}+t_{[1],2}+r_{[1],2},$$

$$C_{[2]}=\max\{C_{[1]}+s_{[2],2}, C_{[1]}-t_{[1],2}-r_{[1],2}+s_{[2],1}+t_{[2],1}+r_{[1],1}\} + t_{[2],2}+r_{[2],2}$$

$$C_{[j]}=\max\{C_{[j-1]}+s_{[j],2}, C_{[j-1]}-t_{[j-1],2}-r_{[j-1],2}+s_{[j],1} + t_{[j],1}+r_{[j-1],1}\}+t_{[j],2}+r_{[j],2}$$

where $C_{[0]}=t_{[0],2}=r_{[0],2}=r_{[0],1}=0$.

$$\begin{aligned}M(S)&=C_{[n]}\\ &=\max\{C_{[n-1]}+s_{[n],2}, C_{[n-1]}-t_{[n-1],2}-r_{[n-1],2}+s_{[n],1}+t_{[n],1} + r_{[n-1],1}\}+t_{[n],2}+r_{[n],2}\\ &=C_{[n-1]}+\max\{s_{[n],2}, s_{[n],1}+t_{[n],1}-t_{[n-1],2}-r_{[n-1],2}+r_{[n-1],1}\} + t_{[n],2}+r_{[n],2}\\ &=\sum_{j=1}^{n}\max\{s_{[j],2}, s_{[j],1}+t_{[j],1}-t_{[j-1],2}-r_{[j-1],2}+r_{[j-1],1}\} + \sum_{j=1}^{n}(t_{[j],2}+r_{[j],2})\\ &=\sum_{j=1}^{n}\max\{0,(s_{[j],1}+t_{[j],1}-s_{[j],2})-(t_{[j-1],2}+r_{[j-1],2}-r_{[j-1],1})\} + \sum_{j=1}^{n}(t_{[j],2}+r_{[j],2}+s_{[j],2}+r_{[j-1],1})\end{aligned}$$

We define $p_{[j],1}=s_{[j],1}+t_{[j],1}-s_{[j],2}$,

$$p_{[j],2}=t_{[j],2}+r_{[j],2}-r_{[j],1}$$

**Theorem 1.** In a sequence where jobs $J_i$ and $J_j$ are adjacent, job $j$ should precede job $i$ for minimizing makespan, assuming $J_j$ should be processed at position $\tau$, then $J_i$ be processed at position $\tau+1$, if the following conditions hold

(1) $p_{i,1}\le p_{i,2}$, (2) $p_{j,1}\le p_{j,2}$, (3) $p_{i,1}\le p_{j,1}$, (4) $p_{j,1}\le p_{[\tau-1],2}$, (5) $p_{j,2}\le p_{i,2}$.

**Proof**. Consider two sequences $S_1$ and $S_2$, $S_1$ has job $J_i$ in position $\tau$ and job $J_j$ in position $\tau+1$ while $S_2$ has job $J_j$ in position $\tau$ and job $J_i$ in position $\tau+1$. The two sequences $S_1$ and $S_2$ have the same job in other positions.

Let $C_{[\tau-1]}(S_1)=C_{[\tau-1]}(S_2)=k$. We have the following equations for the two sequences.

$$M(S_1) = k+\max\{0, p_{i,1}-p_{[\tau-1],2}\}+r_{[\tau-1],2}+s_{i,2}+t_{i,2}+r_{i,2}$$
$$+\max\{0, p_{j,1}-p_{i,2}\}+r_{i,2}+s_{j,2}+t_{j,2}+r_{j,2}$$
$$+\max(0, p_{[\tau+2],1}-p_{j,2})+r_{j,2}+s_{[\tau+2],2}+t_{[\tau+2],2}+r_{[\tau+2],2}$$
$$+\sum_{p=\tau+3}^{n}\max\{0, p_{[k],1}-p_{[k-1],2}\}$$
$$+\sum_{p=\tau+3}^{n}\max\{s_{[p],2}+r_{[p-1],1}+t_{[p],2}+r_{[p],2}\}$$
$$M(S_2) = k+\max\{0, p_{j,1}-p_{[\tau-1],2}\}+r_{[\tau-1],2}+s_{j,2}+t_{j,2}+r_{j,2}$$
$$+\max\{0, p_{i,1}-p_{j,2}\}+r_{j,2}+s_{i,2}+t_{i,2}+r_{i,2}$$
$$+\max\{0, p_{[\tau+2],1}-p_{i,2}\}+r_{i,2}+s_{[\tau+2],2}+t_{[\tau+2],2}+r_{[\tau+2],2}$$
$$+\sum_{p=\tau+3}^{n}\max\{0, p_{[k],1}-p_{[k-1],2}\}$$
$$+\sum_{p=\tau+3}^{n}\max\{s_{[p],2}+r_{[p-1],1}+t_{[p],2}+r_{[p],2}\}$$

Following conditions (1)-(3), we obtain

$$p_{i,1}\le p_{j,2}\le p_{[\tau-1],2} \quad (3)$$

From conditions (4) and equation (3), it is obvious that

$$\max\{0, p_{j,1}-p_{[\tau-1],2}\}=\max\{0, p_{i,1}-p_{[\tau-1],2}\}=0 \quad (4)$$

Following equation (3), we know $p_{i,1}-p_{j,2}\le 0$.

So $\max\{0, p_{i,1}-p_{j,2}\}=0\le\max\{0, p_{j,1}-p_{i,2}\}$ (5)

Also if condition (5) hold, then

$$\max\{0, p_{[\tau+2],1}-p_{i,2}\}\le\max\{0, p_{[\tau+2],1}-p_{j,2}\} \quad (6)$$

From equations (4)-(6), it's obvious that

$$M(S_2)\le M(S_1).$$

Hence, sequence $S_2$ is better than sequence $S_1$, namely $J_j$ should precede $J_i$ in any schedule.

**Theorem 2**. In a sequence where jobs $J_i$ and $J_j$ are adjacent, job $j$ should precede job $i$ for minimizing makespan, assuming $J_j$ should be processed at position $\tau$, then $J_i$ be processed at position $\tau+1$, if the following conditions hold

(1) $p_{i,1}\ge p_{i,2}$, (2) $p_{j,1}\ge p_{j,2}$, (3) $p_{i,2}\ge p_{j,2}$, (4) $p_{j,1}\le p_{[\tau-1],2}$, (5) $p_{i,1}\le p_{[\tau-1],2}$ (6) $p_{i,1}-p_{j,2}\le p_{j,1}-p_{i,2}$.

**Proof.** The proof of this theorem is similar to that of Theorem 1.

We will use the two theorems in the subsequent section as part of a heuristic algorithm. The first three conditions in the two theorems satisfy the classification and sorting of jobs in Johnson rule [9]. We first use Johnson's algorithm to sort jobs with $p_{i,1}$ and $p_{i,2}$ as the processing time in two stages separately. Then use the two theorems to adjust jobs order in the sorted sequence.

## IV. Proposed Heuristic Algorithms

In this section, we provide two heuristics to find the near optimal solutions of the considered problem. Sule has reduced the two-machine flowshop with setup, processing and removal times separated to the classical two-machine flowshop by regarding $p_{i,1}$, $p_{i,2}$ as the processing time of job $i$ in the reduced problem. Then the problem can be resolved by Johnson's algorithm. The heuristic 1 ($H_1$) use the same method to sort the jobs to be processed in the first phase, and the second phase of the heuristic is to assign the jobs in the obtained sequence one by one using assignment rule described in section 2. The heuristic 2 ($H_2$) improves the $H_1$ by using the Theory 1 and Theorem 2.

### A. Heuristic 1

Phase 1: Sequencing the jobs

Step 1. Let $s_{i,2}$, $t_{i,2}$ and $r_{i,2}$ ($i=1,2,...,n$) be divided by the machine number $m$ in the second stage. Then $p_{i,1}$ and $p_{i,2}$ can be computed as following

$p_{i,1}=s_{i,1}+t_{i,1}-s_{i,2}/m$, $p_{i,2}=t_{i,2}/m+r_{i,2}/m-r_{i,1}$.

Step 2. Find the jobs with $p_{i,1}<p_{i,2}$ and forms the set of jobs *U*, $U=\{i \mid p_{i,1}<p_{i,2}\}$

Step 3. Sort the jobs in *U* in non-descending order of $p_{i,1}$.

Step 4: Find the jobs with $p_{i1}\ge p_{i,2}$ and forms the set of jobs *V*, $V=\{i \mid p_{i,1}\ge p_{i,2}\}$

Step 5: Sort the jobs in *V* in non-ascending order of $p_{i,2}$.

Step 6: A schedule is obtained by combined the two sorted set into one sequence, the jobs of the set *U* is put precede the jobs of the set *V*.

Phase 2. Assignment of jobs

Step 7: Let *S*=([1], [2],…, [n]) be the schedule obtained after step 6, for assigning the job on position $i$, compute the set $\{[A_2([i-1])+s_{[i],1}+t_{[i],1}]-[B_{k,2}(l)+s_{[i],2}] \mid k=1,..,m\}$ according to equation (1). If more than one unit in the set are not greater than zero, choose one of the according machines arbitrary as the machine the selected job to be processed on in stage 2. If there is no any unit in the set less than or equal to zero, choose the machine which according unit is the minimal in the set.

Step 8: Accept the final assignment and compute the makespan using equation (2). The obtained schedule is an approximate solution to the problem.

*B. Heuristic 2*

Phase 1: Sequencing the jobs

Step 1. Let $s_{i,2}$, $t_{i,2}$ and $r_{i,2}$ ($i = 1,2,...,n$) be divided by the machine number $m$ in the second stage. Then $p_{i,1}$ and $p_{i,2}$ can be computed as following

$p_{i,1} = s_{i,1} + t_{i,1} - s_{i,2} / m$, $p_{i,2} = t_{i,2} / m + r_{i,2} / m - r_{i,1}$.

Step 2. Find the jobs with $p_{i,1} < p_{i,2}$ and forms the set of jobs *U*, $U = \{i \mid p_{i,1} < p_{i,2}\}$

Step 3. Sort the jobs in *U* in non-descending order of $p_{i,1}$. Update the sorted sequence *U* using Theorem 1, $p_{[0],2} = 0$.

Step 4: Find the jobs with $p_{i1} \geq p_{i,2}$ and forms the set of jobs *V*, $V = \{i \mid p_{i,1} \geq p_{i,2}\}$

Step 5: Sort the jobs in *V* in non-ascending order of $p_{i,2}$, and let the job in the position 0 of the sorted sequence is the last job in the set *U*. Update the sorted sequence *V* using Theorem 2.

Step 6: A schedule is obtained by combined the two sorted set into one sequence, the jobs of the set *U* is put precede the jobs of the set *V*.

The second phase of $H_2$ is the same as the second phase of $H_1$.

## V. COMPUTATIONAL EXPERIENCE

To evaluate the performance of the proposed heuristic, we randomly generated a large number of instances to compute. It has been proved that the uniform distribution with large data spread provides more different problems to solve (refer to [4] and [5]). We designed the following instance sets for uniform distribution:

$I_1$: $1 \leq t_{i,1}, t_{i,2} \leq 20$, $1 \leq s_{i,1}, r_{i,1}, s_{i,2}, r_{i,2} \leq 20k$

$I_2: 1 \leq t_{i,1}, \leq 20$, $1 \leq s_{i,1}, r_{i,1} \leq 20k$;

$1 \leq t_{i,2} \leq 20m, 1 \leq s_{i,2}, r_{i,2} \leq 20mk$

$I_3: 1 \leq t_{i,1}, \leq 50$, $1 \leq s_{i,1}, r_{i,1} \leq 50k$;

$1 \leq t_{i,2} \leq 50m, 1 \leq s_{i,2}, r_{i,2} \leq 50mk$

where $m$ is the number of the machines in stage 2, $k$ is the ratio of setup and removal times to processing time ($s_{i,j} / t_{i,j}$, $r_{i,j} / t_{i,j}$).

For in each instance, we select 100 problems randomly to calculate the percentage deviation of the makespan of the schedule generated by the proposed heuristics from its lower bound. The lower bound of each problem is estimated as

$$LB = \max\{\sum_{i=1}^{n}(s_{i,1} + t_{i,1} + r_{i,1}) - \max_{i=1}^{n} r_{i,1} + \min_{i=1}^{n}(t_{i,2} + r_{i,2}),$$
$$\frac{1}{m}\sum_{i=1}^{n}(s_{i,2} + t_{i,2} + r_{i,2}) - \frac{1}{m}\max_{i=1}^{n} r_{i,2} + \min_{i=1}^{n}(s_{i,1} + t_{i,1})\}$$

The following statistics were collected:

Avg.: Average percentage deviation of the heuristic makespan from its lower bound.

TABLE I.
PERFORMANCE EVALUATION OF THE PROPOSED HEURISTICS

| I | n | m | k | $H_1$ Avg. | $H_1$ Max. | $H_2$ Avg. | $H_2$ Max. | Aimp. |
|---|---|---|---|---|---|---|---|---|
| $I_1$ | 20 | 2 | 0.1 | 1.99 | 7.28 | 1.29 | 4.60 | 35.1 |
| $I_2$ | 20 | 2 | 0.1 | 16.25 | 28.16 | 13.41 | 21.09 | 17.5 |
| $I_3$ | 20 | 2 | 0.1 | 18.20 | 37.10 | 15.08 | 31.41 | 17.1 |
| $I_1$ | 20 | 2 | 0.5 | 1.69 | 4.11 | 1.52 | 3.53 | 10.2 |
| $I_2$ | 20 | 2 | 0.5 | 15.92 | 27.87 | 14.23 | 22.35 | 10.6 |
| $I_3$ | 20 | 2 | 0.5 | 18.27 | 26.82 | 16.45 | 24.74 | 9.9 |
| $I_1$ | 20 | 2 | 1 | 2.25 | 5.13 | 2.19 | 4.54 | 2.8 |
| $I_2$ | 20 | 2 | 1 | 18.64 | 31.33 | 18.53 | 29.69 | 4.8 |
| $I_3$ | 20 | 2 | 1 | 19.21 | 33.01 | 18.26 | 29.92 | 3.9 |
| $I_1$ | 40 | 2 | 0.5 | 1.29 | 4.61 | 1.01 | 3.41 | 21.7 |
| $I_2$ | 40 | 2 | 0.5 | 17.58 | 24.13 | 15.27 | 21.49 | 13.1 |
| $I_3$ | 40 | 2 | 0.5 | 19.22 | 28.56 | 17.09 | 25.46 | 9.5 |
| $I_1$ | 80 | 2 | 0.5 | 1.48 | 4.14 | 1.08 | 2.94 | 27.0 |
| $I_2$ | 80 | 2 | 0.5 | 19.04 | 25.62 | 16.47 | 24.10 | 13. |
| $I_3$ | 80 | 2 | 0.5 | 20.20 | 25.20 | 18.19 | 23.31 | 9.9 |
| $I_1$ | 150 | 2 | 0.5 | 1.54 | 3.26 | 1.09 | 2.65 | 29.2 |
| $I_2$ | 150 | 2 | 0.5 | 18.92 | 23.84 | 16.14 | 21.38 | 14.6 |
| $I_3$ | 150 | 2 | 0.5 | 20.84 | 24.87 | 18.61 | 22.65 | 10.7 |

Max.: Maximum percentage deviation of the heuristic makespan from its lower bound.

Aimp.: The average improvement percentage of $H_2$ to $H_1$.

As shown in Table 1, for each instance the average performance is of $H_2$ better than $H_1$. It is proved that the inclusion Theorem 1 and Theorem 2 as part of the heuristic can improve the Sule's algorithm greatly in no-wait environment. For the instances of $I_1$ where the setup, processing and removal times in both centers are from the same distribution, both of the two heuristics have excellent performance. This because in this distribution the production ability of stage 2 is greater than stage 1, and the jobs can be processed on machine of stage 1 with little or no idle times. For the same $m$ and $k$, the average improvement slightly increases as the number of jobs increase. Further, we can observe that the ratio of setup and removal times to processing time $k$ is smaller, and the improvement of $H_2$ to $H_1$ is larger. In the real world, generally, the setup and removal times are less than processing time greatly, so the heuristic 2 has obviously advantage.

## VI. Conclusions

This paper has discussed the two-stage hybrid no-wait flowshop with separated setup and removal times. Since this problem is NP-complete, there is no known polynomial time algorithm can solve it. We consider one of the approximate methods solving the problem in two phases: sequencing jobs as in pure no-wait flowshop and assignment jobs to machines. We propose two theorems for sequencing procedure. The heuristic algorithm including the theorems has been shown can solve the problems effectively with low computational complexity.

It is possible to develop the similar algorithm for the reverse problem with identical parallel machines in stage 1 and only one machine in stage 2. Further, the algorithm proposed in this paper can be generalized to deal with the two-stage no-wait hybrid flowshop scheduling problem with more than one parallel machine in both stages. And the theorems proposed in this paper can be used in two-stage no-wait pure flowshop scheduling problem.

## References


[1] T. A. Aldowaisan, "A new heuristic and dominance relations for no-wait flowshops with setup times," *Computers and Operations Research*, vol. 28, no. 6, pp. 563-584, May 2001.

[2] T. A. Aldowaisan, and A. Allahverdi, "Total flowtime in no-wait flowshops with separated setup times," *Computers and Operations Research*, vol. 25, no. 12, pp. 1025-1031, Dec. 1998.

[3] A. Allahverdi, "Scheduling in stochastic flowshops with independent setup," processing and removal times. *Computers and Operations Research*, vol. 24, no. 11, pp. 995-960, Nov. 1997.

[4] H. G. Campbell, R. A. Dudek, and M. L. Smith, "A heuristic algorithm for the n-job, m-machine sequencing problem," *Management Science*, vol. 16, no. 10, pp. 630-637, Jun. 1970.

[5] D. G. Dannenbring, "An evaluation of flowshop sequencing heuristics," *Management Science*, vol. 23, no. 11, pp. 1174-1182, Jul. 1977.

[6] J. N. D. Gupta, V. A. Strusevich, and C. Zwaneveld, "Two-stage no-wait scheduling models with setup and removal times," *Computers and Operations Research*, vol. 24, no. 11, pp. 1025-1031, Nov. 1997.

[7] N. G. Hall, and C. A. Sriskandarajah, "A survey of machine scheduling problems with blocking and no-wait in process," *Operations Research*, vol. 44, no. 3, pp. 510-525, May/Jun 1996.

[8] J. N. D. Gupta, and E. A. Tunc, "Scheduling a two-stage hybrid flowshop with separable setup and removal times," *European Journal of Operational Research*, vol. 77, pp. 415-426, 1994.

[9] S. M. Johnson, "Optimal two and three stage production schedules with setup times included," *Naval research logistics quarterly*, vol. 1, pp. 61-68, 1954.

[10] M. S. Salvador, "A solution of a special class of flow shop scheduling problems," in *Proc. of the symposium on the theory of scheduling and its applications*, Springer-Verlag, Berlin, 1973, pp. 83-91.

[11] C. Sriskandarajah, and P. Ladet, "Some no-wait shops scheduling problems: complexity aspect," *European Journal of Operational Research*, vol. 24, no. 3, pp. 424-445, Mar. 1986.

[12] C. Sriskandarajah, and P. Sethis, "Scheduling algorithms for flexible flowshops: Worst and average case performance," *European Journal of Operational Research*, vol. 43, no. 2, pp. 143-160. Nov. 1989.

[13] C. Sriskandarajah, "Performance of scheduling algorithms for no-wait flowshops with parallel machines," *European Journal of Operational Research*, vol. 70, no. 3, pp. 365-378, Nov. 1993.

[14] D. R. Sule, K. Y. Huang, "Sequency on two and three machines with setup, processing and removal times separated," *International Journal of Production Research*, vol. 21, no. 5, pp. 723-732, Sep./Oct. 1983.

[15] Z. Liu, J. Xie, "A heuristic for two-stage no-wait hybrid flowshop scheduling with a single machine in either stage," *Tsinghua Science and Technology*, vol. 8, no. 1, pp. 43-48, Feb. 2003.

[16] A. Allahverdi, T. Aldowaisan, "Minimizing total completion time in a no-wait flowshop with sequence-dependent additive changeover times," *Journal of the Operational Research Society*, vol. 52, pp. 449-462, 2001.

[17] A. Allahverdi, T. Aldowaisan, "No-wait flowshops with bicriteria of makespan and total completion time," *Journal of the Operational Research Society*, vol. 53, no. 9, pp. 1004-1015, 2002.