

$\frac{3}{2}$ -Approximation for Two-machine No-wait Flowshop Scheduling with Availability Constraints

T.C. Edwin Cheng^{1,*} and Zhaohui Liu^{1,2}

¹Department of Management, The Hong Kong Polytechnic University

Kowloon, Hong Kong, China

²Department of Mathematics, East China University of Science and Technology

Shanghai 200237, China

July 20, 2003

Abstract

We consider in this paper the two-machine no-wait flowshop problem in which each machine may have an unavailable interval. We present $\frac{3}{2}$ -approximation algorithms for the problem when the unavailable interval is imposed on only one machine, or the unavailable intervals on the two machines overlap. These algorithms improve on existing results.

Keywords: scheduling, approximation algorithm

*Corresponding author.

1 Introduction

In the two-machine no-wait flowshop problem, each job has to be processed on each machine for a period subject to the constraint that the processing on machine 2 follows the processing on machine 1 without waiting. In this paper, we consider the two-machine no-wait flowshop problem in which each machine may have an availability constraint, i.e., an interval during which the machine is unavailable for processing. Due to the no-wait constraint, the processing of any job cannot be interrupted by the unavailable intervals. Our objective is to minimize the makespan, i.e., the completion time of the last job.

Although the classical two-machine no-wait flowshop problem is polynomially solvable (see Gilmore and Gomory [3] and Hall and Sriskandarajah [4]), the problem in which only one machine has an availability constraint is NP-hard (see Espinouse et al. [2]). Wang and Cheng [5] provided $\frac{5}{3}$ -approximation algorithms for problems with one availability constraint. We will give two improved algorithms for them. Also, we will study the problem in which machine 1 and machine 2 have overlapping unavailable intervals. All of our algorithms have a worst-case performance bound of $3/2$. In another paper (see Cheng and Liu [1]), we have designed a polynomial-time approximation scheme for these problems. The approximation scheme is interesting only in theory since its complexity contains a huge coefficient whose value depends on the accuracy desired. In comparison, the approximation algorithms presented in this paper are more efficient in practice. Moreover, we use different techniques to construct these algorithms.

2 Notation and preliminaries

We first introduce the notation to be used in this paper.

M_1, M_2 : machine 1 and machine 2;

$J = \{1, 2, \dots, n\}$: the set of jobs to be processed;

(a_j, b_j) : an alternative notation of job j , where a_j and b_j denote its processing time on M_1 and M_2 , respectively;

s_i, t_i : M_i ($i = 1, 2$) is unavailable from s_i to t_i , where $0 \leq s_i \leq t_i$;

$d_i = t_i - s_i$: the length of the unavailable interval on M_i ;

$\sigma_{GG}(I)$: the schedule without availability constraints produced by Gilmore and Gomory's algorithm for some job set I ;

$C_{GG}(I)$: the makespan of $\sigma_{GG}(I)$;

$\sigma_{GG}(I, k)$: the schedule without availability constraints produced by Gilmore and Gomory's algorithm for some job set I given $k \in I$ is scheduled as the last job;

$C_{GG}(I, k)$: the makespan of $\sigma_{GG}(I, k)$;

σ : the schedule with given availability constraints produced by our approximation algorithm for J ;

C^* : the optimal makespan for J with given availability constraints.

Note that $\min_{k \in J} C_{GG}(J, k) = C_{GG}(J) \leq C^*$.

The makespan of a schedule (j_1, j_2, \dots, j_n) for the classical two-machine no-wait flowshop problem is

$$a_{j_1} + \sum_{i=1}^{n-1} \max\{a_{j_{i+1}} - b_{j_i}, 0\} + \sum_{i=1}^n b_{j_i}. \quad (1)$$

If k is fixed as the last job, then $j_n = k$ and the problem of minimizing (1) reduces to the traveling salesman problem with n nodes and the cost functions

$$\begin{aligned} c_{kj} &= a_j, \\ c_{ij} &= \max\{a_j - b_i, 0\} \quad (i \neq k). \end{aligned}$$

Let $A_j = a_j$ ($j = 1, 2, \dots, n$), $B_i = b_i$ ($i \neq k$) and $B_k = 0$, and introduce functions $f(x) = 1$ and $g(x) = 0$. Then,

$$c_{ij} = \begin{cases} \int_{B_i}^{A_j} f(x) dx & \text{if } A_j \geq B_i, \\ \int_{A_j}^{B_i} g(x) dx & \text{if } A_j < B_i. \end{cases}$$

Gilmore and Gomory [3] gave an $O(n \log n)$ algorithm for the traveling salesman problem with such cost functions, i.e., an $O(n \log n)$ algorithm to generate $\sigma_{GG}(J, k)$.

Instead of fixing a job as the last job, we introduce an auxiliary job with zero processing time on both machines to act as the last job. So, $\sigma_{GG}(J)$ can also be obtained in $O(n \log n)$ time.

3 M_1 has an unavailable interval

In this section, we present a $\frac{3}{2}$ -approximation algorithm for the two-machine no-wait flowshop problem with an availability constraint on M_1 . The algorithm works on the following main ideas:

- (i) try to find a good schedule in which the availability constraint is inactive (see Step 1);
- (ii) relax the availability constraint to obtain a super-optimal schedule, and then move some jobs from the beginning to the end or vice versa to meet the availability constraint (see Steps 3 and 4);
- (iii) schedule optimally some critical job and its adjacent jobs in $\sigma_{GG}(J)$, and schedule the other jobs according to Gilmore and Gomory's algorithm (see Steps 2 and 5).

Algorithm 1

Step 1: Construct $\sigma_{GG}(J, k)$ for each $k \in J$. If there are some schedules with the completion time of all jobs on M_1 no more than s_1 , then let $\sigma_{GG}(J, k_1)$ be the shortest one of such schedules, else go to Step 3. If $C_{GG}(J, k_1) = C_{GG}(J)$ or $C_{GG}(J, k_1) \leq t_1$, then let $\sigma = \sigma_{GG}(J, k_1)$ and stop.

If $C_{GG}(J, k_1) = C_{GG}(J)$, σ sure is optimal. If $C_{GG}(J, k_1) \leq t_1$, then $\sigma_{GG}(J, k_1)$ has the minimum makespan among all schedules that complete no later than t_1 since the schedules must complete on M_1 no later than s_1 . σ is optimal too.

Step 2: Let k_1 be followed by k_2 in $\sigma_{GG}(J)$. Let σ_1 and σ_2 denote the schedules with the unavailable interval $[s_1, t_1]$ determined by the job sequences $(k_1, k_2, \sigma_{GG}(J \setminus \{k_1, k_2\}))$ and $(k_2, k_1, \sigma_{GG}(J \setminus \{k_1, k_2\}))$, respectively. Then, σ is given by the shortest one among $\sigma_{GG}(J, k_1)$, σ_1 and σ_2 . Stop.

When Step 2 is performed, $C_{GG}(J, k_1) > C_{GG}(J)$ holds. Then, k_1 is not the last job in $\sigma_{GG}(J)$ and k_2 exists. Moreover, since $C_{GG}(J, k_1) > t_1$, we have $C^* > t_1$. In the case of $C_{GG}(J, k_1) > \frac{3}{2}C^*$, it holds that $b_{k_1} > \frac{1}{2}C^*$ and

$$C_{GG}(J \setminus \{k_1, k_2\}) \leq C_{GG}(J) - b_{k_1} < \frac{1}{2}C^*.$$

So the shortest one among $\sigma_{GG}(J, k_1)$, σ_1 and σ_2 has makespan no more than $\frac{3}{2}C^*$.

Step 3: Construct $\sigma_{GG}(I_k)$ for each $k \in J$, where $I_k = (J \setminus \{k\}) \cup \{(d_1 + a_k, b_k)\}$. Let $C_{GG}(I_{k'}) = \min\{C_{GG}(I_k) \mid k \in J\}$ and s'_1 denote the start time of $(d_1 + a_{k'}, b_{k'})$ in $\sigma_{GG}(I_{k'})$.

When Step 3 needs to be performed, there must be some jobs starting after the unavailable interval $[s_1, t_1]$ in any feasible schedule. Then, $(d_1 + a_k, b_k)$ can be viewed as a relaxation of the unavailable interval and its immediately succeeding job k . Then, $C_{GG}(I_{k'})$ is a lower bound for C^* .

Step 4: Convert $\sigma_{GG}(I_{k'})$ into a schedule for J with the unavailable interval $[s_1, t_1]$ as follows. If $s'_1 > s_1$, then shift the jobs starting in $[0, s'_1 - s_1)$ to the end of the schedule, else shift the jobs starting in $[C_{GG}(I_{k'}) - s_1 + s'_1, C_{GG}(I_{k'}))$ to the beginning. Replace $(d_1 + a_{k'}, b_{k'})$ by the unavailable interval $[s_1, t_1]$ and the succeeding job k' . Let σ_0 denote the resulting schedule and k_0 denote the last job in σ_0 .

In the case of $s'_1 > s_1$, replacing $(d_1 + a_{k'}, b_{k'})$ by $[s_1, t_1]$ and k' makes the jobs originally starting in $[0, s'_1 - s_1)$ start before time zero. Shifting them to the end increases the makespan by no more than the length of the last job before time zero. In the case of $s'_1 < s_1$, replacing $(d_1 + a_{k'}, b_{k'})$ by $[s_1, t_1]$ and k' brings some idleness at the beginning. Shifting the jobs starting in $[C_{GG}(I_{k'}) - s_1 + s'_1, C_{GG}(I_{k'}))$ to the beginning reduces the idle time to no more than the length of the last job starting before $C_{GG}(I_{k'}) - s_1 + s'_1$. Thus, the makespan of σ_0 exceeds $C_{GG}(I_{k'})$ by at most $a_{k_0} + b_{k_0}$.

Step 5: Let K be the set including k_0 and its adjacent jobs in $\sigma_{GG}(J)$. Let σ be the shortest one among σ_0 and the schedules with the unavailable interval $[s_1, t_1]$ determined by the job sequences in the form of $(K, \sigma_{GG}(J \setminus K))$. Stop.

Since $|K| \leq 3$, there are at most six sequences in the form of $(K, \sigma_{GG}(J \setminus K))$. If $a_{k_0} + b_{k_0} \leq \frac{1}{2}C^*$, then σ_0 has makespan no more than $\frac{3}{2}C^*$. If $a_{k_0} + b_{k_0} > \frac{1}{2}C^*$, then

$$C_{GG}(J \setminus K) \leq C_{GG}(J) - a_{k_0} - b_{k_0} < \frac{1}{2}C^*,$$

and the shortest schedule determined by $(K, \sigma_{GG}(J \setminus K))$ has makespan no more than $\frac{3}{2}C^*$.

Theorem 1 *σ obtained by Algorithm 1 is a $\frac{3}{2}$ -approximation for the two-machine no-wait flowshop problem with an availability constraint on M_1 .*

The complexity of Algorithm 1 is dominated by Steps 1 and 3, each of which needs to call Gilmore and Gomory's algorithm n times. Since the complexity of Gilmore and Gomory's algorithm is $O(n \log n)$, the complexity of Algorithm 1 is $O(n^2 \log n)$.

4 M_2 has an unavailable interval

In this section, we give a similar approximation algorithm for the two-machine no-wait flowshop problem with an availability constraint on M_2 .

Algorithm 2

Step 1: Construct $\sigma_{GG}(J)$. If $C_{GG}(J) \leq s_2$, then let $\sigma = \sigma_{GG}(J)$ and stop.

σ obtained in Step 1 is optimal.

Step 2: Construct $\sigma_{GG}(I_k)$ for each $k \in J$, where $I_k = (J \setminus \{k\}) \cup \{(\max\{a_k - d_2, 0\}, b_k)\}$. Let $C_{GG}(I_{k'}) = \min\{C_{GG}(I_k) \mid k \in J\}$ and s'_2 denote the start time of $(\max\{a_{k'} - d_2, 0\}, b_{k'})$ on M_2 in $\sigma_{GG}(I_{k'})$.

When Step 2 is performed, there must be some jobs processed on M_2 after the unavailable interval $[s_2, t_2]$ in any feasible schedule. Since an overlapping interval of length d_2 on M_1 and M_2 is not reckoned in $C_{GG}(I_{k'})$ when $(\max\{a_{k'} - d_2, 0\}, b_{k'})$ acts as the relaxation of the unavailable interval and its immediately succeeding job k' , $C_{GG}(I_{k'}) + d_2$ is a lower bound for C^* .

Step 3: Convert $\sigma_{GG}(I_{k'})$ into a schedule for J with the unavailable interval $[s_2, t_2]$ as follows. If $s'_2 > s_2$, then shift the jobs before $(\max\{a_{k'} - d_2, 0\}, b_{k'})$ and starting in $[0, s'_2 - s_2)$ to the end of the schedule, else shift the jobs after $(\max\{a_{k'} - d_2, 0\}, b_{k'})$ and starting in $[C_{GG}(I_{k'}) - s_2 + s'_2, C_{GG}(I_{k'}))$ to the beginning. Replace $(\max\{a_{k'} - d_2, 0\}, b_{k'})$ by the unavailable interval $[s_2, t_2]$ and the succeeding job k' . Let σ_0 denote the resulting schedule and k_0 denote the last job in σ_0 .

The makespan of σ_0 exceeds $C_{GG}(I_{k'}) + d_2$ by at most $a_{k_0} + b_{k_0}$, where the reason is similar to that of Step 4 of Algorithm 1.

Step 4: Let K be the set including k_0 and its adjacent jobs in $\sigma_{GG}(J)$. Let σ be the shortest one among σ_0 and the schedules with the unavailable interval $[s_2, t_2]$ determined by the job sequences in the form of $(K, \sigma_{GG}(J \setminus K))$. Stop.

As in Section 3, we can prove that either σ_0 or the shortest schedule determined by $(K, \sigma_{GG}(J \setminus K))$ has makespan no more than $\frac{3}{2}C^*$.

Theorem 2 σ obtained by Algorithm 2 is a $\frac{3}{2}$ -approximation for the two-machine no-wait flowshop problem with an availability constraint on M_2 .

The complexity of Algorithm 2 is dominated by Step 2, so it is $O(n^2 \log n)$.

5 M_1 and M_2 have overlapping unavailable intervals

Finally, we consider the problem in which M_1 and M_2 have overlapping unavailable intervals, i.e., $[s_1, t_1] \cap [s_2, t_2] \neq \emptyset$. Note that unless $s_1 = t_2$, there is no job starting before the unavailable interval on M_1 and completing after the unavailable interval on M_2 .

Algorithm 3

Step 1: Construct $\sigma_{GG}(J, k)$ for each $k \in J$. If there exist some $\sigma_{GG}(J, k)$ with $C_{GG}(J, k) \leq \min\{s_1 + b_k, s_2\}$, then let σ be the shortest one of such schedules and stop.

σ obtained in Step 1 is optimal.

Step 2 (executed only if $s_1 = t_2$ and $\sum_{j=1}^n a_j \leq s_1$): Construct $\sigma_{GG}(H_k, (\max\{a_k - d_2, 0\}, 0))$ for each $k \in J$, where $H_k = (J \setminus \{k\}) \cup \{(\max\{a_k - d_2, 0\}, 0)\}$. If there exist some k such that $C_{GG}(H_k, (\max\{a_k - d_2, 0\}, 0)) \leq s_2$, then let k^* be one of such indices with the minimum b_{k^*} , else go to Step 3. If $b_{k^*} \leq d_1$, then let σ be the schedule obtained from $\sigma_{GG}(H_{k^*}, (\max\{a_{k^*} - d_2, 0\}, 0))$ by replacing $(\max\{a_{k^*} - d_2, 0\}, 0)$ by job k^* and the given unavailable intervals, and stop.

Step 2 constructs all schedules in which the last job starts before $[s_1, t_1]$ on M_1 and completes after $[s_2, t_2]$ on M_2 , and lets the shortest one be σ if its makespan is no more than t_1 (i.e., $b_{k^*} \leq d_1$). Thus, σ obtained in Step 2 is optimal and has makespan $s_1 + b_{k^*}$. If the algorithm does not stop here, then $C^* > \max\{t_1, t_2\}$.

Step 3 (executed only if $s_1 = t_2$): Construct $\sigma_{GG}(I_k)$ for each $k \in J$ with $a_k \leq s_1$, where $I_k = (J \setminus \{k\}) \cup \{(\max\{a_k - d_2, 0\}, \max\{b_k - d_1, 0\})\}$. Let $C_{GG}(I_{k'}) = \min\{C_{GG}(I_k) \mid k \in J \text{ and } a_k \leq s_1\}$.

If there is a job starting before the unavailable interval on M_1 and completing after the unavailable interval on M_2 in an optimal schedule, then $C_{GG}(I_{k'}) + d_1 + d_2 \leq C^*$, where $d_1 + d_2$ makes up for the overlapping interval of length $d_1 + d_2$ on M_1 and M_2 which is not reckoned in $C_{GG}(I_{k'})$.

Step 4: Construct $\sigma_{GG}(I_0)$ for $I_0 = J \cup \{(\max\{s_2 - s_1, 0\}, \max\{t_2 - t_1, 0\})\}$.

If there is no job starting before the unavailable interval on M_1 and completing after the unavailable interval on M_2 in an optimal schedule, then $C_{GG}(I_0) + t_1 - s_2 \leq C^*$, where $t_1 - s_2$ makes up for the overlapping interval of length $t_1 - s_2$ on M_1 and M_2 which is not reckoned in $C_{GG}(I_0)$.

Step 5: Convert $\sigma_{GG}(I_0)$ and $\sigma_{GG}(I_{k'})$ (if available) into schedules for J with the given availability constraints as in Algorithms 1 and 2. Let σ_0 denote the shorter one of the resulting schedules and k_0 denote the last job in σ_0 .

The makespan of σ_0 exceeds C^* by at most $a_{k_0} + b_{k_0}$.

Step 6: Let K be the set including k_0 and its adjacent jobs in $\sigma_{GG}(J)$. Let σ be the shortest one among σ_0 and those schedules with the given availability constraints determined by the job sequences in the form of $(K, \sigma_{GG}(J \setminus K))$. Stop.

As in Section 3, we can prove that either σ_0 or the shortest schedule determined by $(K, \sigma_{GG}(J \setminus K))$ has makespan no more than $\frac{3}{2}C^*$.

Theorem 3 *σ obtained by Algorithm 3 is a $\frac{3}{2}$ -approximation for the two-machine no-wait flowshop problem in which M_1 and M_2 have overlapping unavailable intervals.*

The complexity of Algorithm 3 is dominated by Steps 1 \sim 3, so it is $O(n^2 \log n)$. But if $s_2 \leq s_1 < t_2$, the complexity reduces to $O(n \log n)$ since it suffices to compute $\sigma_{GG}(J)$ in Step 1, Steps 2 and 3 are not performed, and the complexity of other steps is no more than $O(n \log n)$.

Acknowledgment

This research was supported in part by The Hong Kong Polytechnic University under grant number G-YW59. The second author was also supported by the National Natural Science Foundation of China under grant number 10101007.

References

- [1] T.C.E. Cheng, Z. Liu, Approximability of two-machine no-wait flowshop scheduling with availability constraints, *Operations Research Letters* 31 (2003) 319–322.
- [2] M.L. Espinouse, P. Formanowicz, B. Penz, Minimizing the makespan in the two-machine no-wait flow-shop with limited machine availability, *Computers & Industrial Engineering* 37 (1999) 497–500.
- [3] P.C. Gilmore, R.E. Gomory, Sequencing a one-state variable machine: a solvable case of the traveling salesman problem, *Operations Research* 12 (1964) 655–679.

- [4] N.G. Hall, C. Sriskandarajah, A survey of machine scheduling problems with blocking and no-wait in process, *Operations Research* 44 (1996) 510–525.
- [5] G. Wang, T.C.E. Cheng, Heuristics for two-machine no-wait flowshop scheduling with an availability constraint, *Information Processing Letters* 80 (2001) 305–309.