# Approximability of Two-Machine No-Wait Flowshop Scheduling with Availability Constraints

T.C. Edwin Cheng[1,*] and Zhaohui Liu[1,2]

[1]Department of Management, The Hong Kong Polytechnic University

Kowloon, Hong Kong, China

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

Shanghai 200237, China

October 5, 2002

## Abstract

We consider in this paper the two-machine no-wait flowshop scheduling problem in which each machine may have an unavailable interval. We present a polynomial time approximation scheme for the problem when the unavailable interval is imposed on only one machine, or the unavailable intervals on the two machines overlap.

*Keywords*: scheduling, approximation scheme

*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 scheduling 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 [2] and Hall and Sriskandarajah [3]), the problem with an unavailable interval becomes NP-hard, and the problem with two separate unavailable intervals has no polynomial time approximation with constant performance bound unless $P = NP$ (see Espinouse et al. [1]). Wang and Cheng [5] provided 5/3-approximation algorithms for the problem with an unavailable interval. In this paper, we present a polynomial time approximation scheme (PTAS) for the problem in which machine 1 and machine 2 have overlapping unavailable intervals or only one machine has an unavailable interval.

# 2 Notation and preliminaries

We first introduce some notation to be used throughout this paper.

$M_1, M_2$ : machine 1 and machine 2;

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

$a_j$ , $b_j$ : the processing time of job $j$ on $M_1$ and $M_2$;

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

$\sigma_{GG}(I)$ : 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 $\sigma_{GG}(I,k)$;

$\sigma_A$ : the schedule generated by our approximation scheme for $J$;

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

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

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

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 $A_j = a_j$ $(j = 1, 2, \ldots, n)$, $B_i = b_i$ $(i \neq k)$ 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 [2] 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. Also, we note the following relation

$$\frac{1}{2}\sum_{j=1}^{n}(a_j + b_j) \leq C_{GG}(J) = \min_{k \in J} C_{GG}(J,k) \leq C^*.$$

# 3 An approximation scheme

In this section, we present an approximation scheme for the two-machine no-wait flowshop scheduling problem in which the unavailable intervals $[s_1, t_1]$ and $[s_2, t_2]$ satisfy one of the following conditions: (1) $s_1 < t_2$ and $s_2 \leq t_1$; (2) $s_1 = +\infty$; (3) $s_2 = +\infty$. Condition (1) states that $M_1$ and $M_2$ have overlapping unavailable intervals and implies there is no job having its first operation processed before $[s_1, t_1]$ on $M_1$ and its second operation processed after $[s_2, t_2]$ on $M_2$. Conditions (2) and (3) respectively imply that $M_1$ and $M_2$ have no availability constraint.

In the approximation scheme, we first try to find an optimal schedule in which all jobs are completed before the unavailable intervals. Failing this, borrowing an idea from Sevastianov and Woeginger [4], we partition the job set $J$ into three subsets: $L$, $S$ and $T$, which consist of large jobs, small jobs and tiny jobs, respectively, and then schedule each subset in one or two consecutive segments without availability constraints. The following is the approximation scheme.

*Step 1.* Construct $\sigma_{GG}(J,k)$ for each $k \in J$. If there exist some $\sigma_{GG}(J,k)$ with $C_{GG}(J,k) \le \min\{s_1 + b_k, s_2, t_1\}$, then let $\sigma_A$ be the shortest one of such schedules and stop.

*Step 2.* Let $\epsilon > 0$, and

$$S(k) = \left\{ j \in J \,|\, \epsilon^k C_{GG}(J) > a_j + b_j > \epsilon^{k+1} C_{GG}(J) \right\},$$

for $k = 1, 2, \ldots, \lceil 2/\epsilon \rceil$. Determine $k^*$ such that

$$\sum_{j \in S(k^*)} (a_j + b_j) \le \epsilon C_{GG}(J)\,. \tag{2}$$

Let

$$\begin{aligned} L &= \left\{ j \in J \,|\, a_j + b_j \ge \epsilon^{k^*} C_{GG}(J) \right\}, \\ S &= S(k^*)\,, \\ T &= \left\{ j \in J \,|\, a_j + b_j \le \epsilon^{k^*+1} C_{GG}(J) \right\}. \end{aligned}$$

*Step 3.* Construct $\sigma_{GG}(S)$ and $\sigma_{GG}(T)$.

*Step 4.* For each pair $(L_1, k)$ with $k \in L_1 \subseteq L$, do

(i) Construct $\sigma_{GG}(L_1, k)$. If $C_{GG}(L_1, k) \le \min\{s_1 + b_k, s_2\}$, then go to (ii), else turn to another $(L_1, k)$.

(ii) Divide $\sigma_{GG}(T)$ into two segments $\sigma_1$ and $\sigma_2$ such that the front segment $\sigma_1$ can be placed into the gap at the beginning of $\sigma_{GG}(L_1, k)$ and has the most jobs (push the jobs in $L_1$ backward to reduce the gap before the unavailable intervals when necessary). Put $\sigma_1$ at the beginning of $\sigma_{GG}(L_1, k)$.

(iii) Schedule the jobs in $L \setminus L_1$ to follow $\sigma_{GG}(L_1, k)$ and the unavailable intervals according to Gilmore and Gomory's algorithm. This can be done by reversing time, exchanging machine names, and creating an auxiliary job that simulates the end of $\sigma_{GG}(L_1, k)$ and the unavailable intervals and is scheduled as the last job.

(iv) Put $\sigma_2$ after $L \setminus L_1$.

(v) Put $\sigma_{GG}(S)$ after $\sigma_2$. Let $\sigma_{L_1,k}$ denote the resulting schedule.

*Step 5.* Let $\sigma_A$ be the shortest one of all $\sigma_{L_1,k}$ obtained in Step 4.

# 4 Analysis of the approximation scheme

$\sigma_A$ obtained in Step 1 is optimal since it has the minimum makespan among all schedules in which all jobs are completed before the unavailable intervals. If the algorithm enters Step 2, there must be some jobs completed after the unavailable intervals in an optimal schedule, i.e., $C^* > t$, where $t = \max\{t_1, t_2\}$ if both $t_1$ and $t_2$ are limited, $t = \min\{t_1, t_2\}$ otherwise.

Since $\sum_{j=1}^{n}(a_j + b_j) \leq 2C_{GG}(J)$ and all $S(k)$ are disjoint, $k^*$ satisfying (2) exists; otherwise, it holds that

$$\sum_{k=1}^{\lceil 2/\epsilon \rceil} \sum_{j \in S(k)} (a_j + b_j) > \left\lceil \frac{2}{\epsilon} \right\rceil \cdot \epsilon C_{GG}(J) \geq 2C_{GG}(J)\,,$$

a contradiction. It follows from (2) that $C_{GG}(S) \leq \epsilon C^*$. Since $C^* > t$, appending $\sigma_{GG}(S)$ to the end of a $1 + O(\epsilon)$-approximation for $L \cup T$ leads to a $1 + O(\epsilon)$-approximation for $J$.

Since

$$|L|\epsilon^{k^*} C_{GG}(J) \leq \sum_{j \in L}(a_j + b_j) \leq 2C_{GG}(J)\,,$$

it holds that $|L| \leq 2/\epsilon^{k^*} \leq 2\epsilon^{-\lceil 2/\epsilon \rceil}$. We next prove a lemma.

**Lemma 1** *The problem of scheduling $L \cup T$ with the unavailable intervals $[s_1, t_1]$ and $[s_2, t_2]$ has an approximation solution such that the tiny jobs in $T$ are processed first or last and the makespan is at most $(1 + 4\epsilon)C^*$.*

**Proof** Consider an optimal schedule $\sigma$ for $L \cup T$ with the unavailable intervals $[s_1, t_1]$ and $[s_2, t_2]$. Let the tiny jobs in $T$ be partitioned into $m$ segments by the large jobs in $L$ and the unavailable intervals in $\sigma$, and the jobs in the first $l$ ($0 \leq l \leq m$) segments be started before the unavailable intervals. For $i = 1, 2, \ldots, m$, let $j_i$ be the first job and $j'_i$ the last job in the $i$th segment. Note that it is possible that $j_i = j'_i$. We transform $\sigma$ by two steps:

(1) shift the tiny jobs started before the unavailable intervals to the beginning and the tiny jobs started after the unavailable intervals to the end (without changing their relative order), and then push all jobs toward the unavailable intervals to compress the machine idleness (at this stage, it is allowable that some tiny jobs are scheduled before time zero);

(2) shift the tiny jobs started before time zero to the end.

After the first step, the increase in the length of the part started after the unavailable intervals is bounded by

$$a_{j_{l+1}} + \sum_{i=l+1}^{m-1} \max\left\{b_{j'_i}, a_{j_{i+1}}\right\} + b_{j'_m} \leq (m-l)\epsilon^{k^*+1} C_{GG}(J)\,,$$

and the length of the part before time zero is bounded by

$$a_{j_1} + \sum_{i=1}^{l-1} \max\left\{b_{j'_i}, a_{j_{i+1}}\right\} + b_{j'_l} \leq l\epsilon^{k^*+1} C_{GG}(J)\,.$$

Then, the makespan of the resulting schedule after the second step exceeds $t$ or the original makspan of $\sigma$ by at most

$$m\epsilon^{k^*+1} C_{GG}(J) \leq (|L|+2)\epsilon^{k^*+1} C_{GG}(J) \leq 4\epsilon C^*\,.$$

This completes the proof. □

Let $L'_1 \subseteq L$ be the set of large jobs started before the unavailable intervals in an approximation solution $\sigma'$ for $L \cup T$ satisfying the requirement in Lemma 1 and $k'$ be the last job in $L'_1$. Let $\sigma''$ be the schedule obtained in (i)-(iv) of Step 4 for $L \cup T$ when $(L_1, k) = (L'_1, k')$. Note that the length of the part consisting of the large jobs in $\sigma''$ does not exceed the length of the corresponding part in $\sigma'$. The makespan of $\sigma''$ exceeds $t$ or the makespan of $\sigma'$ by the length of at most three tiny jobs. Then, it is at most $(1 + O(\epsilon))C^*$, where the constant in the $O$-notation does not depend on $\epsilon$. Consequently, $\sigma_{L'_1, k'}$ is a $1 + O(\epsilon)$-approximation for the job set $J$.

The complexity of the approximation scheme is dominated by Step 4, which needs to call Gilmore and Gomory's algorithm $O(2^{|L|}n)$ times, so it is $O(2^{|L|}n^2 \log n)$. Since $|L| \leq 2\epsilon^{-\lceil 2/\epsilon \rceil}$, the approximation scheme is a PTAS. We have thus established the following theorem.

**Theorem 1** *The two-machine no-wait flowshop scheduling problem with the unavailable intervals $[s_1, t_1]$ and $[s_2, t_2]$ has a PTAS if $s_1 < t_2$ and $s_2 \leq t_1$, or $s_1 = +\infty$, or $s_2 = +\infty$.*

As a corollary of Theorem 1, we can also prove the following theorem.

**Theorem 2** *The two-machine no-wait flowshop scheduling problem with the unavailable intervals* $[s_1, t_1]$ *and* $[s_2, t_2]$ *has a PTAS if* $s_1 = t_2$*.*

**Proof** Suppose that no jobs have zero processing time on both machines; otherwise, they can be scheduled at the beginning with no cost. In the case of $s_1 = t_2$, it is possible that some job $i$ or some two jobs $j$ and $k$ with $b_j = a_k = 0$ are processed before $[s_1, t_1]$ on $M_1$ and processed after $[s_2, t_2]$ on $M_2$. We apply the approximation scheme in Section 3 to the job set $J$ with the unavailable intervals $[s_1, t_1]$ and $[s_2, t_2]$, all $J \setminus \{i\}$ with the unavailable intervals $[s_1 - a_i, t_1]$ and $[s_2, t_2 + b_i]$, and all $J \setminus \{j, k\}$ with the unavailable intervals $[s_1 - a_j, t_1]$ and $[s_2, t_2 + b_k]$. The shortest one of the resulting schedules is a $1 + O(\epsilon)$-approximation for the problem. □


# 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] 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.

[2] 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.

[3] N.G. Hall, C. Sriskandarajah, A survey of machine scheduling problems with blocking and no-wait in process, Operations Research 44 (1996) 510–525.

[4] S.V. Sevastianov, G.J. Woeginger, Makespan minimization in open shop: a polynomial time approximation scheme, Mathematical Programming 82 (1998) 191–198.

[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.