Complexity of Minimize Total Weighted Average Completion Time Scheduling Problems with Release Times

SHI Ling\(^1\), XU Chun\(^1\), CHENG Xue-guang\(^2\)

1. Department of Mathematics
Hubei University for Nationalities, Enshi 445000, China
shiling59@126.com

2. School of Mathematics and Statistics
Wuhan University, Wuhan 430072, China

Abstract

The paper considers the problem of scheduling \( n \) jobs in a two-machine flow-shop to minimize the weighted sum of completion times. Between the completion of an operation and the beginning of the next operation of the same job, there is a time lag, which wherever to it as the transportation delays. All transportation delays have to be done by a single robot, which can perform at most one transportation at a time. And release dates \( r_j \) may be given for the jobs. New complexity results are derived for special case. For a new heuristic algorithm, the worst-case performance is 3/2, and the bound is tight.

Mathematics Subject Classification: 90B35

Keywords: flow-shop scheduling problem; transportation delays; single robot; release times; complexity
1 Introduction

In the flow-shop scheduling model, we are given $m$ machines $M_1, M_2, \ldots, M_m$, where $m \geq 0$ and $n$ jobs $J_1, J_2, \ldots, J_n$. Each job $J_j$ consists of a chain $(O_{1,j}, O_{2,j}, \ldots, O_{m,j})$ of operations, and $Q_{i,j} (i = 1, 2, \ldots, m; j = 1, 2, \ldots, n)$ is to be processed on machine $M_i$ for $p_{i,j}$ time units. Each machine can process at most one job at a time, and each job can be processed on at most one machine at a time. No preemption is allowed, i.e., one started, any operation can not be interrupted before it is completed. In this paper, we assume that there is a known time lag between the completion of an operation and the beginning of the next operation of the same job. We refer to this lag as the transportation delays, i.e., if operation $Q_{j,k}$ is processed on machine $M_k$ and afterwards $Q_{k+1,j}$ on machine $M_{k+1}$, a transportation delay $t_{j,k}$ arises. We consider the case where all transportation is done by a single transportation robot $R$, which can only handle one job at a time. Thus, conflicts between transportation may arise and a job may have to wait for the robot before its transportation. Furthermore, release dates $r_j$ may be given for the jobs, weights $w_j$ may be used in the objective function. All values $p_{i,j}$, $t_{j,k}$, $r_j$ and $w_j$ are supposed to be non-negative integers.

The objective is to determine a feasible schedule, which minimizes the weighted sum of completion times $\sum_{j=1}^{n} w_j C_j$, where $C_j$ is the finishing time of the last operation $Q_{m,j}$ of job $J_j$. Using the three-field notation scheme for scheduling problem introduced in [1], we denote the $m$-machine flow-shop scheduling problem with release times, transportation delays and a single robot by $Fm, Rt|p_{i,j}; t_{j,k}; r_j; \sum_{j=1}^{n} w_j C_j$.

If we have only $m = 2$ machines, the robot always transports from $M_1$ to $M_2$. Therefore, the index $k$ in the notation $t_{j,k}$ is dropped and the transportation delays are
denoted by \( t_j \). We denote this problem by \( F2, R | p_{i,j}; t_j; r_j | L_j \). Throughout the paper we also will consider the special cases where all operations \( O_{i,j} \) have unit processing times \( p_{i,j} = 1 \). We defined these problems as \( F2, R | p_{i,j} = 1; t_j; r_j | L_j \). The \( F2, R | p_{i,j} = 1; t_j | C_{\text{max}} \) problem has maximal polynomial solvable \([2]\). The \( F2 | p_{i,j} = 1; t_j; r_j | L_j \sum C_j \) problem, and the \( F2 | p_{i,j} = 1; t_j; r_j | L_j \sum w_j C_j \) problem are \( NP \)-hard in the strong sense, too \([3-7]\).

In this paper, we discuss the \( F2, R | p_{i,j} = 1; t_j; r_j | L_j \sum w_j C_j \) problem and proof these problems is \( NP \)-hard in the strong sense. For the \( F2, R | p_{i,j} = 1; t_j; r_j | L_j \sum w_j C_j \) problem we give a heuristic algorithm and the worst-case performance is 3/2, and the bound is tight.

2 Complexity of the \( F2, R | p_{i,j} = 1; t_j; r_j | L_j \sum w_j C_j \) problem

In this section, we consider problem in which we have three machines \( M_1, M_2 \), one robot \( R \), and \( n \) jobs \( J \) with processing times \( p_{1,j} \) and \( p_{2,j} \) on machine \( M_1 \) and \( M_2 \). Without loss of generality, we assume that \( 0 = r_1 \leq r_2 \leq \ldots \leq r_n \). We may restrict the search for an optimal solution to permutation plans, since for problem \( F3 | C_{\text{max}} \) has an optimal permutation plan always exists \([5]\).

We now derive an expression for the makespan when the sequences \( \sigma \) and \( \tau \) in which the jobs are executed by \( M_1 \) and \( M_2 \) are given. Let \( C(\sigma, \tau) \) denote the
minimal makespan of such a schedule for the \( F_2,R\| p_{i,j} = 1; t_j \sum_{j=1}^{n} w_j C_j \) problem.

**Lemma 1** Consider the \( F_2,R\| p_{i,j} = 1; t_j, r_j \sum_{j=1}^{n} w_j C_j \) problem with processing times \( p_{i,j} \) and transportation delays \( t_j \), where \( i = 1,2 \) and \( j = 1,2,...,n \). Then

\[
wC(\sigma, \tau) = \max_{1 \leq k \leq m} \{ r_j + \sum_{j \in \sigma^{-1}(k)} P_{i,j} + t_k + \sum_{j \in \tau^{-1}(k)} P_{2,j} \}
\]

Where \( \sigma^{-1}(k) \) and \( \tau^{-1}(k) \) denote the positions of job \( k \) in sequence \( \sigma \) and \( \tau \), respectively.

**Proof:** The proof is same as Wensi Yu’ theorem [8].

**Theorem 1:** The \( F_2,R\| p_{i,j} = 1; t_j, r_j \sum_{j=1}^{n} w_j C_j \) problem is \( NP \)-hard in the strong sense.

**Proof:** We prove the \( F_2,R\| p_{i,j} = 1; t_j, r_j \sum_{j=1}^{n} w_j C_j \) problem is \( NP \)-hard in the strong sense through a reduction from the 3–Partition problem. The 3–Partition problem is then stated as:

3–Partition: Given a set of positive integers \( X = \{x_1, x_2, ..., x_{3m}\} \), and a positive integer \( b \) with:

\[
\sum_{j=1}^{3m} x_j = mb, \quad \frac{b}{4} < x_j < \frac{b}{2}, \forall j = 1,2,...,3m
\]

Decide whether there exists a partition of \( X \) into \( m \) disjoint 3-element subset \( \{X_1, X_2, ..., X_m\} \) such that \( \sum_{x_j \in X_i} x_j = b \) \((i = 1,2,...,m)\)

(2.3)

There are \( n = 2m + mb \) jobs split into four groups: the \( P \)-jobs (partition jobs) denoted by \( P_j (j = 1,2,...,3m) \), the \( X \)-jobs(first auxiliary jobs) denoted by \( X_j (j = 1,2,...,m) \), the \( Y \)-jobs (last auxiliary jobs) denoted by \( Y_j (j = 1,2,...,m) \), and the \( Z \)-jobs (zero delays jobs) denoted by \( Z_j (j = 1,2,...,m(b-3)) \). Their transportation delays, release times, and weights are given by the formulas: (1)
Partition jobs, or $P$ -jobs with:  

\[ t_{p_j} = x_j, \quad w_{p_j} = b, \quad r_{p_j} = 0 \quad (j = 1, 2, \ldots, 3m) \]

(2) First auxiliary jobs, or $X$ -jobs with:

\[ t_{x_j} = bj + m + 1, \quad w_{x_j} = (m + 1 - j)H, \quad r_{x_j} = (j - 1)(b + 2) \quad (j = 1, 2, \ldots, m) \]

(3) Last auxiliary jobs, or $Y$ -jobs with:

\[ t_{y_j} = (m - j)b + m + 1, \quad w_{y_j} = (m + 1 - j), \quad r_{y_j} = j(b + 1) + (j + 1) \quad (j = 1, 2, \ldots, m) \]

(4) Zero delays jobs or $Z$ -jobs with:  

\[ t_{z_j} = 0, \quad w_{z_j} = b, \quad r_{z_j} = 0 \quad (j = 1, 2, \ldots, m(b - 3)) \]

Where  

\[ H = W(P, Z) + W(Y) \]

Define the threshold  

\[ y = W(X) + W(P, Z), \]

where

\[ W(X) = \sum_{j=1}^{m} (m + 1 - j)H(j - 1)(b + 2), \quad W(Y) = \sum_{j=1}^{m} (m + 1 - j)(j(b + 1) + (j - 1)), \]

\[ W(P, Z) = \frac{m}{2}(mb + m + b + 5). \]

And the corresponding decision problem is: Is there a schedule $S$ with makespan $C(S)$ not greater than $y = W(X) + W(Y) + W(P, Z)$?

Assume that the answer to 3–Partition is ‘yes’, Let \( \{X_1, X_2, \ldots, X_m\} \) be a partition satisfying (2.3), where

\[ X_i = \{x_{\xi(i)}, x_{\eta(i)}, x_{\zeta(i)}\} \quad (i = 1, 2, \ldots, m). \]

Suppose that the sets \( G_j (j = 1, 2, \ldots, m) \) form a solution to the 3–Partition problem. A schedule $S^*$ can be found in the following way. Let $\xi_i, \eta_i, \zeta_i$ denote an arbitrary permutation of the $P$ -jobs whose indices belong to set $G_j$, and let $Z(k)$ denote an arbitrary permutation of $x_k - 1$ $Z$ -jobs. We construct for each $j = 1, 2, \ldots, m$ a schedule $S_j$ consisting of jobs $\xi_i, \eta_i, \zeta_i$, and $Z(\xi_i), Z(\eta_i), Z(\zeta_i)$ as shown in Figure 1.
The whole schedule $S^*$ between $[r_j, r_{j+1}]$ is obtained by scheduling $A$-jobs and subschedule $S_j (j = 1, 2, ..., m)$ shown in Figure 2.

Then we define a permutation $\sigma$ shown in Figure 1. Obviously, this permutation $\sigma$ fulfills $wC(\sigma) \leq y$.

Conversely, assume that the flow-shop scheduling problem has a solution $\sigma$ with $wC(\sigma) \leq y$.

By setting $k = 1, j = n, t_j = 0$ in (2.1), we get for all permutation $\sigma$:

$$wC(\sigma) \geq \sum_{\lambda=1}^{m+1} w_{1,\sigma_{\lambda}} p_{1,\sigma_{\lambda}} + \sum_{\lambda=1}^{m} w_{2,\tau_{\lambda}} p_{2,\tau_{\lambda}} = W(X) + W(P, Z)$$
Thus, for a permutation $\sigma$ with $wC(\sigma) = y$. We may conclude that:

1. machine $M_1$ processed jobs in the interval $[0, m(b+1) + m]$, without idle times;
2. machine $M_2$ processed jobs in the interval $[2, 3 + m(b+1) + (m-1)]$, without idle times;
3. robot $R$ transport jobs in the interval $[1, 2 + m(b+1) + (m-1)]$, without idle times.

Suppose an instance $I$ of 3-partition, and an instance $J$ of the $\sum_{j=1}^{n} w_j C_{wrtpRF}$ problem. Assume that the answer to the $F2, Rl | p_{i,j} = 1; t_j; r_j | \sum_{j=1}^{n} w_j C_{j}$ problem is ‘yes’. An instance $J$ of the $F2, Rl | p_{i,j} = 1; t_j; r_j | \sum_{j=1}^{n} w_j C_{j}$ problem does not have a 3-partition.

Consider ant schedule $\sigma$ of Jon two machines. We want to prove that the value of the objective function for $\sigma$ exceeds $y$.

We consider easy cases first. If some $P$-jobs is completed after time $3 + jb$, then

$$\sum_{j=1}^{m} w_j C_j > b(3 + jb) + \sum_{l=1, l \neq j}^{m} b(lb) + \sum_{j=1}^{m} [(m + 1 - j)(3 + j(b+1) + (j-2))]$$

$$+ \sum_{j=1}^{m} [(m + 1 - j)(3 + j(b+1) + (j-1))]$$

$$+ \sum_{j=1}^{m} [(m + 1 - j)(3 + j(b+1) + (j-2))] > y$$

If some $X$-jobs is completed after time $3 + j(b+1)$, then

$$\sum_{j=1}^{m} w_j C_j > (m+1-j)H(3 + j(b+1) + (j-1)) + \sum_{l=1, l \neq j}^{m} (m+1-j)H(3 + j(b+1) + (j-1))$$

$$+ \sum_{j=1}^{m} (3 + jb) + \sum_{j=1}^{m} (m + 1 - j)(3 + j(b+1) + (j-2)) > y$$
If some $Y$-jobs is completed after time $3 + (j + 1)(b + 1)$, then
\[
\sum_{j=1}^{m} w_j C_j > (m+1 - j)(3 + j(b+1) + (j-2)) + \sum_{l=1, l \neq j}^{m} [(m+1 - j)(3 + l(b+1) + (l-2))]
\]
\[
\sum_{j=1}^{m} (3 + jb)b + \sum_{j=1}^{m} [(m+1 - j)H(3 + j(b+1) + (j-1))] > y
\]

Suppose now that each $X$-job is completed no later than at time $3 + j(b+1)$, each $Z$-job is completed no later at time $2 + j(b+1)$, and each $Y$-job is completed no later than at time $3 + (j + 1)(b+1)$. Since I does not have a 3-patition, some machine is idle in machine 1 or machines 2, which contradict (1)~(3). Summarizing completing the proof of the claim.

3 A heuristic algorithm for the $F2, R||p_{i,j} = 1; t_j; r_j \sum_{j=1}^{n} w_j C_j$ problem

3.1 Algorithm
Step 1 Calculate $w_i t_i = \min\{w_i t_i \mid i \leq n\} (1 \leq l \leq n)$;

Step 2 put the jobs in order of non-decreasing ratios $w_i t_i (1 \leq i \leq n)$, that is

\[
w_1 t_1 \leq w_2 t_2 \leq \ldots \leq w_{j-1} t_{j-1} \leq w_j t_j \leq w_{j+1} t_{j+1} \leq \ldots \leq w_n t_n
\]
If there exists $w_i t_i = w_j t_j (i \neq j)$, then arrange the jobs in order of minimal index first;

Step 3 Schedule the jobs in accordance with the above until finish the whole jobs.

3.2 The worst-case rate
Let $S_j$ denote the start time of job $J_j$; $I_{1,i}$ denote the idle time of job $J_i$ on machine 1;

\[
S^*_{H} = \sum (w_j C_j)^+ \text{ denote the optimum of the } F2, R||p_{i,j} = 1, t_j, r_j \sum_{j=1}^{n} w_j C_j \text{ problem.}
\]

Theorem 2 For the $F2, R||p_{i,j} = 1, t_j, r_j \sum_{j=1}^{n} w_j C_j$ problem, supports $S_H$ is a schedule
Scheduling problems with release times

according to Algorithm 1, then \( \frac{C_{\max}(S_H)}{C_{\max}(S^*)} \leq \frac{3}{2} \), and the bound is tight.

**Proof** \( w_jC_j = w_j(r_j + p_{1,j} + t_j + p_{2,j} + S_j) = w_j(r_j + p_{1,j} + t_j + p_{2,j} + \sum_{i=1}^{j}(p_{1,i} + I_{i,j})) \)

\[
= w_j(r_j + p_{1,j} + p_{2,j}) + w_j t_j + w_j \left( \sum_{i=1}^{j}(p_{1,i} + I_{i,j}) \right)
\]

\[
\leq w_j(r_j + p_{1,j} + p_{2,j}) + w_j t_{j+1} + w_j \left( \sum_{i=1}^{j}(p_{1,i} + I_{i,j}) \right)
\]

\[
\sum_{j=1}^{n} w_j C_j \leq \sum_{j=1}^{n} \left[ w_j(r_j + p_{1,j} + p_{2,j}) + w_j t_{j+1} + w_j \left( \sum_{i=1}^{j}(p_{1,i} + I_{i,j}) \right) \right] \quad (3.1)
\]

Similarly, we have

\[
\sum_{j=2}^{n+1} w_{j-1} C_{j-1} \leq \sum_{j=2}^{n+1} \left[ w_{j-1}(r_{j-1} + p_{1,j-1} + p_{2,j-1}) + w_j t_j + w_{j-1} \left( \sum_{i=1}^{j-1}(p_{1,i} + I_{i,j-1}) \right) \right] \quad (3.2)
\]

Combine (3.1) and (3.2), we have

\[
\sum_{j=1}^{n} w_j C_j \leq \left( \sum_{j=1}^{n} w_j(r_j + p_{1,j} + t_j + p_{2,j}) \right) + \left( \sum_{j=2}^{n+1} w_{j-1}(r_{j-1} + p_{1,j-1} + t_{j-1} + p_{2,j-1}) \right) + \sum_{j=1}^{n} \left( p_{1,j} + I_{i,j} \right) \leq 3 \sum (w_j C_j)^*
\]

That is \( \frac{C_{\max}(S_H)}{C_{\max}(S^*)} \leq \frac{3}{2} \).

To prove the bound is tight, introduce the following example

1. \( w_1 = 1, t_1 = 7, r_1 = 1 \); 2. \( w_j = 0, t_j = 0, r_j = 0 (j = 2, 3, ..., 7) \)

As show in Figure 3 and 4.

\[
\begin{array}{cccccccc}
J_1 & J_2 & J_3 & J_4 & J_5 & J_6 & J_7 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
J_2 & J_3 & J_4 & J_5 & J_6 & J_7 & J_1 \\
\end{array}
\]

Fig.3 \( C_{\max}^*(H) \)
$r_2 = 0$

So we have $\frac{C_{\text{max}}(S_H)}{C_{\text{max}}(S_{H}^{*})} = \frac{15}{10} = \frac{3}{2}$, the bound is tight.

Acknowledgements. This work is supported by Education Department of Hubei Province (No.B20082907).

References


Received: June, 2009