Complexity Results for Flow-shop Scheduling Problems with Transportation Delays and a Single Robot

Shi Ling\textsuperscript{1} and Cheng Xue-guang\textsuperscript{2,\ast}

Abstract

The paper considers the problem of scheduling $n$ jobs in a two-machine flow-shop to minimize the makespan. Between the completion of an operation and the beginning of the next operation of the same job, there is a time lag, which we refer 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. New complexity results are derived for special case.

Mathematics Subject Classification : 90B35

Keywords: Flow-shop Scheduling problem, Complexity Results, Transportation Delay, Single Robot

\textsuperscript{1} Department of Mathematics, Hubei University for Nationalities, Enshi 445000, China, e-mail: Shiling59@126.com
\textsuperscript{2} School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China, e-mail: chengxueguang6011@msn.com
\ast Corresponding author.

Article Info: Revised : February 15, 2011. Published online : May 31, 2011
1 Introduction

A flow-shop scheduling problem with transportation delays and a single robot can be formulated as follows. We are given $m$ machines $M_1, M_2, \ldots, M_m$ and $n$ jobs $J_1, J_2, \ldots, J_n$.

Each job $J_j$ consists of $m$ operations $Q_{i,j}$, where $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$. These operations have to be processed in the order $Q_{1,j} \rightarrow Q_{2,j} \rightarrow \cdots \rightarrow Q_{m,j}$.

Operation $Q_{i,j}$ has to be processed on machine $M_i$ without preemption for $p_{i,j}$ time units. Each machine can only process one operation at a time. 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 $t_{j,k}$. All transportation is done by a single 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. All values $p_{i,j}$ and $t_{j,k}$ are supposed to be non-negative integers.

The objective is to determine a feasible schedule, which minimizes the makespan $C_{\text{max}} = \max_{j=1}^{n} C_j$, 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 problems introduced in [4], we denote this problem by $Fm,R|p_{i,j};t_{j,k}|C_{\text{max}}$. 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 $t_j$. If two operations $Q_{1,j}$ and $Q_{2,j}$ have equal processing times $p_{1,j} = p_{2,j} = p_j$, we denote
this problem by $F_2, R\| p_{1,j} = p_{2,j} = p_j; t_j \| C_{\text{max}}$. If the transportation delays may take only two values $T_1, T_2 \quad (T_1 < T_2)$, we have the $F_2, R\| p_{1,j} = p_{2,j} = p_j; t_j \in \{T_1, T_2\} \| C_{\text{max}}$ problem.

The $F_2\| p_{1,j} = p_{2,j} = p_j; t_j \in \{T_1, T_2\} \| C_{\text{max}}$ problem is $\mathcal{NP}$-hard in the strong sense, [5]. J.Hurink and S.Knust discussed the complexity results for the two-machine flow-shop scheduling problem with transportation delays and a single robot and proved the $F_2, R\| p_{1,j} = p_j; t_j \in \{T_1, T_2\} \| C_{\text{max}}$ problem have maximal polynomial solvable, [3]. In this paper, we prove the $F_2, R\| p_{1,j} = p_{2,j} = p_j; t_j \in \{T_1, T_2\} \| C_{\text{max}}$ problem is $\mathcal{NP}$-hard in the strong sense.

2 Complexity of the $F_2, R\| p_{1,j} = p_{2,j} = p_j; t_j \in \{T_1, T_2\} \| C_{\text{max}}$ problem

In this section, we consider problem in which we have two machines $M_1, M_2$, one robot $R$, and $n$ jobs $J_j$ with processing times $p_{1,j}$ and $p_{2,j}$ on machine $M_1$ and $M_2$.

We may restrict the search for an optimal solution to permutation plans, since for problem $F_3\| \| C_{\text{max}}$ has an optimal permutation plan always exists, [1].

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 $F_2, R\| p_{1,j} = p_{2,j} = p_j; t_j \in \{T_1, T_2\} \| C_{\text{max}}$ problem.
Lemma 2.1 [5] Consider the \( F2, R | p_{1,j} = p_{2,j} = p_j; t_j \in \{ T_1, T_2 \} | C_{\text{max}} \) problem with processing times \( p_{i,j} \) and transportation delays \( t_j \), where \( i = 1, 2 \) and \( j = 1, 2, \ldots, n \). Then

\[
C(\sigma, \tau) = \max_{1 \leq k \leq n} \left\{ \sum_{j \sigma^{-1}^{-1}(k)} p_{1,j(\sigma)} + t_k + \sum_{j \tau^{-1}^{-1}(k)} p_{2,j(\tau)} \right\} \tag{2.1}
\]

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

Theorem 2.1 The \( F2, R | p_{1,j} = p_{2,j} = p_j; t_j \in \{ T_1, T_2 \} | C_{\text{max}} \) problem is \( NP \)-hard in the strong sense.

Proof We prove the \( F2, R | p_{1,j} = p_{2,j} = p_j; t_j \in \{ T_1, T_2 \} | C_{\text{max}} \) problem is \( NP \)-hard in the strong sense through a reduction from the 3-Partition problem, which is known to be \( NP \)-hard in the strong sense, [2]. The 3-Partition problem is then stated as:

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

\[
\sum_{j=1}^{3m} x_j = mb, \quad b/4 < x_j < b/2, \forall j = 1, 2, \ldots, 3m \tag{2.2}
\]

Decide whether there exists a partition of \( X \) into \( m \) disjoint 3-element subset \( \{ X_1, X_2, \ldots, X_m \} \) such that

\[
\sum_{x_j \in X_i} x_j = b \quad (i = 1, 2, \ldots, m) \tag{2.3}
\]

Given any instance of the 3-Partition problem, we define the following instance of the \( F2, R | p_{1,j} = p_{2,j} = p_j; t_j \in \{ T_1, T_2 \} | C_{\text{max}} \) problem with two types of jobs:
(1) $3m$ Partition jobs, or P-jobs with:
\[ p_{1,j} = x_j, \quad t_j = 0; \quad p_{2,j} = x_j \quad (j = 1, 2, ..., 3m) \]

(2) $m$ Large jobs, or L-jobs with:
\[ p_{1,j} = 2b, \quad t_j = 2b; \quad p_{2,j} = 2b \quad (j = 3m + 1, 3m + 2, ..., 4m) \]

The threshold $y = 3mb + 3b$ and the corresponding decision problem is: Is there a schedule $S$ with makespan $C(S)$ not greater than $y = 3mb + 3b$?

Assume that the answer to $3 - Partition$ is “yes”, Let $\{X_1, X_2, ..., X_m\}$ be a partition satisfying (2.3), where $X_i = (x_{\xi(i)}, x_{\eta(i)}, x_{\zeta(i)})$ ($i = 1, 2, ..., m$).

We construct for each $j$ consisting of jobs $\xi(j), \eta(j), \zeta(j)$ and jobs $3m + j$ in the order

\[ ((3m + 1); \xi(1), \eta(1), \zeta(1); (3m + 2); \xi(2), \eta(2), \zeta(2); ...; (4m - 1); \xi(m), \eta(m), \zeta(m); 4m) \]

as indicated in Figure 1.

![Gantt chart](image)

Figure 1: Gantt chart for the $F2,R1|p_{1,j} = p_{2,j} = p_j, t_j \in \{T_1, T_2\}|C_{\text{max}}$ problem

Then we define a permutation $\sigma$ shown in Figure 1. Obviously, this permutation $\sigma$ fulfills $C(\sigma) \leq y$. Conversely, assume that the flow-shop scheduling problem has a solution $\sigma$ with $C(\sigma) \leq y$. By setting $k = 1, i = n, t_j = 0$ in (2.1), we get for all
permutation $\sigma$: $C(\sigma) \geq p_{1,\sigma_1} + \sum_{\lambda=1}^{n} p_{2,\sigma_\lambda} = 3b + 3mb = y$.

Thus, for a permutation $\sigma$ with $C(\sigma) = y$. We may conclude that:

1. job $(3m + 1)$ is processed at the first position, since $p_{1,j} > 0$ for $j \neq 0$;
2. job $4m + m$ is processed at the last position, since $p_{2,j} > 0$ for $j \neq m$;
3. machine $M_1$ processed jobs in the interval $[0,3mb]$, without idle times;
4. machine $M_2$ processed jobs in the interval $[3b,3mb+3b]$, without idle times;
5. robot $R$ transport jobs in the interval $[(3i + 2)b_i,(3i + 4)b]$ $(i = 0,1,\ldots,(m-1))$, without idle times.

Without loss of generality, we can assume that the jobs in $\{1,2,\ldots,m-1,m\}$ are processed w.r.t. increasing numbers. Let $X_1 = \{i_1,i_2,\ldots,i_k\}$ be the set of jobs scheduled between job $(3m + 1)$ and job $(3m + 2)$, showing in Figure 2.

![Figure 2: Subscheduling for the $F2,RI|p_{1,j} = p_{2,j} = p_j; t_j \in \{T_1,T_2\}|C_{max}$ problem.](image)

We have $X_1 \neq \Phi$, since otherwise there would be an idle period on the job $(3m + 1)$ and job $(3m + 2)$, which contradicts (3) ~ (5).
In the following we will show that \( k = 3 \), and \( \sum_{x_i \in X_1} x_i = b \) hold. We use the variable \( C_{1,i,j}^\sigma \) denoting the completion time of job \( j \) on machine \( M_i \) in the permutation \( \sigma \).

The values of the variable for the jobs on the set \( X_1 \) are given by:

\[
C_{1,i,j}^\sigma = 2b + \sum_{\lambda=1}^{\mu} p_{1,i,j} < 2b + 2b(\mu + 1) \quad (\mu = 1, 2, ..., k)
\]

If \( k \leq 2 \) holds, we have:

\[
\sum_{\lambda=1}^{k} p_{1,i,j} < k \cdot 2b \leq 2kb + (2 - k)2b = 4b
\]

Then \( C_{1,i}^\sigma = 2b + \sum_{\lambda=1}^{k} p_{1,i,j} < 3b \), and the robot finishes the transportation of job \((3m+1)\) at time \(2b\). Thus, machine \( M_2 \) has an idle time period between jobs \((3m+1)\) and job \((3m+2)\), which contradicts (5);

If \( k \geq 4 \) holds, we have:

\[
\sum_{\lambda=1}^{k} p_{2,i,j} < k \cdot 2b \leq 2bk + (k - 4)2b = 4b(k - 2).
\]

On the other hand, job \((3m+2)\) cannot start on machine \( M_2 \) earlier than time \(2b + kb\), since job \((3m+1)\) have to be transport before. Thus, the time period between the completion time \( C_{2,1}^\sigma = 6b \) for job \((3m+1)\) on machine \( M_2 \) and the starting time of job \((3m+2)\) on machine \( M_1 \) is not completely filled with jobs from \( X_1 \), which contradicts (4); Thus, we must have \( k = 3 \). This implies that job \((3m+1)\) and job \((3m+2)\) transported by robot in the interval \([2b, 3b]\) and \([3b, 4b]\), respectively. Therefore, \(2b + \sum_{x_i \in X_1} p_{i,j} \leq 3b\), that is:

\[
\sum_{x_i \in X_1} p_{i,j} \leq b \quad (2.4)
\]
On the other hand, job \((3m + 1)\) completes on machine \(M_2\) not after \(6b\). Since we have no idle time on machine \(M_2\) in interval \([4b, 6b]\), we must have \(2b + \sum_{i \in X_1} p_{1,i} + \sum_{i \in X_2} p_{2,i} \geq 4b\). Since \(p_{1,j} = p_{2,j} = x_j\), therefore
\[
\sum_{i \in X_1} p_{2,j} \geq b
\]  
(2.5)
Combining (2.4) and (2.5), we have \(\sum_{j \in X} x_j = b\).

Analogously, we show that the remaining sets \(X_2, X_3, \ldots, X_m\) separated by the jobs 1, 2, ..., \(m\) contain 3-element and fulfill \(\sum_{j \in X_j} x_j = b\) for \(j = 1, 2, \ldots, m\). Thus, \(X_1, X_2, \ldots, X_m\) define a solution of 3–Partition.

**Reference**


