Complexity of Two-Machine Scheduling with Bundling Operations and Total Flow-Time Minimization

George Vairaktarakis

Weatherhead School of Management Department of Operations,
Case Western Reserve University 10900
Euclid Avenue Cleveland, OH 44106-7235, U.S.A.
Email: George.Vairaktarakis@case.edu

Abstract

We consider a two-machine scheduling problem in which each job consists of a single operation per machine, and the operations of a job can be performed in parallel. The completion time of a job is the maximum of the completion times of its operations. The objective is to minimize the total flow time of the jobs. In this article, we provide an alternative NP-hardness proof for the problem.

Key words : Scheduling; bundling operations; computational complexity

1. Introduction

We consider a scheduling environment in which each job consists of two components. There are two independent machines, each of which produces its own type of components. A job is completed when the processing of both of its components is finished. This type of scheduling problem is encountered frequently in manufacturing, where finished components are required to be bundled together for final assembly or delivery. For example, in the production of paper products, a given type of product is produced on a specific machine. Orders from different customers usually specify certain quantities of various types of paper products, but an entire order must be shipped to the customer in a single shipment (see Leung *et al.* 2004). See also Lee *et al.* (1993), Sung and Yoon (1998), and Cai and Zhou (2004) for other applications.

Our model is mathematically defined as follows. There are two machines M_1 and M_2, and a given set of n jobs $\{J_1, J_2, \ldots, J_n\}$, where each job J_j has a pair of tasks, namely an a-task and a b-task. The a-task of J_j will be processed by M_1 and requires an uninterrupted processing time of $a_j \geq 0$, while the b-task of J_j will be processed by M_2 and requires an uninterrupted processing time of $b_j \geq 0$ $(j = 1, 2, \ldots, n)$. The completion time of a job is the larger of the completion times of its two tasks. All jobs are available for processing at time 0. The objective is to schedule the a-tasks and b-tasks of the jobs on the machines so as to minimize the total flow time (i.e., the sum of completion times of the jobs). We denote this problem as **P**.

The complexity of **P** has been considered by Wagneur and Sriskandarajah (1993), who presented a strongly NP - hardness proof of the problem. However, Leung *et al.* (2005) have shown that Wagneur and Sriskandarajah's proof is incorrect. Ahmadi *et al.* (2005) and Yang (2005) have proven independently that the two-machine bundling operations problem with an objective of minimizing total job completion time is NP-hard in the strong sense.

A number of researchers have analyzed the computational complexity of other variants of the problem. For example, Sung and Yoon (1998) have shown that the more general problem of minimizing the total weighted completion time is NP- hard in the strong sense. Leung *et al.* (2004) have shown that the extended problem with three or more machines is also strongly NP - hard. However, as mentioned in Leung *et al.* (2005), the complexity of **P** remains open. In this paper, we provide an alternative NP-hardness proof.

2. NP-Hardness

We first provide some important properties of the optimal solution of the problem.

Lemma 1 *There exists an optimal solution for* **P** *in which (i) there is no idle time on either machine, and (ii) the job sequences on both machines are identical.*

Proof: The proof of property (i) is straightforward and is omitted. See Sung and Yoon (1998) for a proof of property (ii).

It suffices to restrict our search to solutions that satisfy the properties stated in Lemma 1. In the remainder of our analysis we will only consider schedules that satisfy these properties.

We will show that **P** is NP - hard by transforming the Even - Odd Partition problem, a known NP - hard problem (Garey *et al.* 1988), into **P**. The Even- Odd Partition problem can be described as follows. Given a set $A = \{1, 2, \ldots, 2m\}$ and a size $x_j \in Z^+$ for each $j \in A$, where $x_1 < x_2 < \cdots < x_{2m}$, does there exist a partition of A into subsets A_1 and A_2 such that $\sum_{k\in A_1} x_k = \sum_{k\in A_2} x_k$, and that each of A_1, A_2 contains exactly one of $2j-1, 2j$ for $j = 1, 2, \ldots, m$? Here, a partition $\{A_1, A_2\}$ is said to be *even-odd* if each of A_1, A_2 contains exactly one of $2j-1, 2j$ for $j = 1, 2, \ldots,$ m. In other words, the Even-Odd Partition problem is to determine an even-odd partition $\{A_1, A_2\}$ of A that satisfies the condition $\sum_{k\in A_1} x_k = \sum_{k\in A_2} x_k$.

Given an instance of the Even-Odd Partition problem, we define the following "large quantities":

$$K = \tfrac{1}{2}\sum_{j=1}^{m}(x_{2j} - x_{2j-1});$$

$$L = \tfrac{1}{2}\sum_{k=1}^{2m} x_k + \tfrac{m}{2}(x_{2m} - x_1);$$

$$R = m(2^m + 1) + (m+1)K.$$

For $j = 1, 2, \ldots, m$, we define the following:

$$y_{2j-1} = x_{2j-1} + \big[(m+1)R + 3\big]2^{m-1}L + 2^{j-1}L;$$

$$y_{2j} = x_{2j} + \big[(m+1)R + 3\big]2^{m-1}L + 2^{j-1}L.$$

In other words, we transform the quantities x_{2j-1} and x_{2j} into y_{2j-1} and y_{2j}, respectively, by adding a large constant to each of them, and such a constant is dependent on j. Clearly, $y_1 < y_2 < \cdots < y_{2m}$. In addition, $\sum_{k\in A_1} x_k = \sum_{k\in A_2} x_k$ if and only if $\sum_{k\in A_1} y_k = \sum_{k\in A_2} y_k = B$ for any even-odd partition $\{A_1, A_2\}$, where $B = \tfrac{1}{2}\sum_{k=1}^{2m} y_k$. Then, we construct the following instance of the scheduling problem:

$$n = 2m + 1;$$

$$a_{2j-1} = (2^{m-1} - 2^{j-1})RL + (m+1)(y_{2j} - y_{2j-1}) \qquad (j = 1, 2, \ldots, m);$$

$$a_{2j} = (2^{m-1} - 2^{j-1})RL \qquad (j = 1, 2, \ldots, m);$$

$$b_k = y_k \qquad (k = 1, 2, \ldots, 2m);$$

$$a_{2m+1} = B + m(y_{2m} - y_1);$$

$$b_{2m+1} = \big[(m-2)2^{m-1} + 1\big]RL + (m+1)K + m(y_{2m} - y_1).$$

In this constructed instance there are $2m$ "regular" jobs $J_1, J_2, \ldots, J_{2m}$ and a "critical" job J_{2m+1}. As will be shown later, this instance is constructed in such a way that in the optimal solution the critical job must be scheduled in the $(m+1)$ st position, separating the regular jobs into two groups with an equal number of jobs. Obviously, this construction can be done in polynomial time. Define a threshold value

$$T = (m+1)(B + b_{2m+1}) + \sum_{j=1}^{m}(m - j + 1)(y_{2j-1} + y_{2j}). \tag{1}$$

We will show that there exists an even-odd partition $\{A_1, A_2\}$ of A such that $\sum_{k \in A_1} y_k = \sum_{k \in A_2} y_k$ if and only if there is a feasible solution to the constructed instance with an objective function value no greater than T. Let $C'_{[k]}$ and $C''_{[k]}$ denote the completion times of the job in the kth position of the job sequence on M_1 and M_2, respectively.

Given a solution to the Even - Odd Partition problem instance, we construct a schedule $\tilde{\sigma}$ of our problem as follows. First, assign J_{2m+1} to the $(m+1)$st position of the job sequence. Then, for $j = 1, 2, \ldots, m$, if $2j - 1 \in A_1$, assign J_{2j-1} to the jth position and J_{2j} to the $(j+m+1)$st position of the job sequence, otherwise assign J_{2j} to the jth position and J_{2j-1} to the $(j + m + 1)$st position of the job sequence. The following lemma implies that $\tilde{\sigma}$ is the desired schedule.

Lemma 2 *The total flow time of jobs in schedule $\tilde{\sigma}$ is equal to T.*

Proof: Note that because $\sum_{k \in A_1} y_k = \sum_{k \in A_2} y_k$, we have

$$\sum_{\substack{j=1,2,\ldots,m \\ \text{s.t. } 2j \in A_2}} y_{2j} + \sum_{\substack{j=1,2,\ldots,m \\ \text{s.t. } 2j-1 \in A_2}} y_{2j-1} = \sum_{\substack{j=1,2,\ldots,m \\ \text{s.t. } 2j-1 \in A_1}} y_{2j-1} + \sum_{\substack{j=1,2,\ldots,m \\ \text{s.t. } 2j \in A_1}} y_{2j},$$

implying that

$$\sum_{\substack{j=1,2,\ldots,m\\ \text{s.t. } 2j-1\in A_1}} (y_{2j} - y_{2j-1}) = \sum_{\substack{j=1,2,\ldots,m\\ \text{s.t. } 2j-1\in A_2}} (y_{2j} - y_{2j-1}),$$

which in turn implies that

$$\sum_{\substack{j=1,2,\ldots,m\\ \text{s.t. } 2j-1\in A_1}} (y_{2j} - y_{2j-1}) = \frac{1}{2}\sum_{j=1}^{m}(y_{2j} - y_{2j-1}) = \frac{1}{2}\sum_{j=1}^{m}(x_{2j} - x_{2j-1}) = K. \tag{2}$$

The completion time of the critical job on M_1 is

$$\begin{aligned}
C'_{[m+1]} &= \sum_{\substack{j=1,2,\ldots,m\\ \text{s.t. } 2j-1\in A_1}} a_{2j-1} + \sum_{\substack{j=1,2,\ldots,m\\ \text{s.t. } 2j\in A_1}} a_{2j} + a_{2m+1}\\
&= \sum_{j=1}^{m}(2^{m-1} - 2^{j-1})RL + (m+1)\sum_{\substack{j=1,2,\ldots,m\\ \text{s.t. } 2j-1\in A_1}} (y_{2j} - y_{2j-1}) + a_{2m+1}\\
&= \big[(m-2)2^{m-1} + 1\big]RL + (m+1)K + a_{2m+1} \qquad \text{(by (2))}\\
&= B + b_{2m+1},
\end{aligned}$$

while the completion time of the same job on M_2 is

$$C''_{[m+1]} = \sum_{k\in A_1} b_k + b_{2m+1} = \sum_{k\in A_1} y_k + b_{2m+1} = B + b_{2m+1}.$$

In other words, $C'_{2m+1} = C''_{2m+1} = B + b_{2m+1}$ (see Figure 1). Note also that $a_j < b_j$ for $j = 1, 2, \ldots, 2m$. Hence, the sum of the job completion times of the schedule is $\sum_{k=1}^{2m+1} C''_{[k]}$. For $j = 1, 2, \ldots, m$,

$$C''_{[j]} = \sum_{\substack{\ell=1,2,\ldots,j\\ \text{s.t. } 2\ell-1\in A_1}} b_{2\ell-1} + \sum_{\substack{\ell=1,2,\ldots,j\\ \text{s.t. } 2\ell-1\in A_2}} b_{2\ell}$$

and

$$C''_{[j+m+1]} = B + b_{2m+1} + \sum_{\substack{\ell=1,2,\ldots,j\\ \text{s.t. } 2\ell-1\in A_1}} b_{2\ell} + \sum_{\substack{\ell=1,2,\ldots,j\\ \text{s.t. } 2\ell-1\in A_2}} b_{2\ell-1},$$

which implies that

$$C''_{[j]} + C''_{[j+m+1]} = B + b_{2m+1} + \sum_{\ell=1}^{j}(b_{2\ell-1} + b_{2\ell}) = B + b_{2m+1} + \sum_{\ell=1}^{j}(y_{2\ell-1} + y_{2\ell}).$$

Therefore,

$$\sum_{k=1}^{2m+1} C''_{[k]} = (m+1)(B + b_{2m+1}) + \sum_{j=1}^{m}\sum_{\ell=1}^{j}(y_{2\ell-1} + y_{2\ell}) = T.$$

Conversely, suppose that there exists a schedule for the constructed instance of our problem with an objective function value of no more than T. Let $\hat{\sigma}$ be an optimal $\hat{\sigma}$. Hence, $Z(\hat{\sigma}) \leq T$. Let $b_{[k]}$ denote the schedule of the constructed instance. Let $Z(\hat{\sigma})$ denote the total flow time of schedule processing time of the b-task in the kth position of the job sequence in schedule $\hat{\sigma}$. We have the following lemma.

Lemma 3 *In schedule $\hat{\sigma}$, the critical job J_{2m+1} must be scheduled in the $(m+1)st$ position.*

Proof: Consider any schedule σ of the constructed instance with r jobs preceding J_{2m+1}, where $r \neq m$. Case 1: $r \geq m+1$. Note that $2^{j-1}RL - (m+1)(y_{2j} - y_{2j-1}) > 0$ for $j = 1, 2, \ldots, m$. Thus, $a_k < 2^{m-1}RL$, for $k = 1, 2, \ldots, 2m$, which implies that $C'_{[r]} < r2^{m-1}RL$. Hence, in this case, schedule σ must satisfy the following:

$$\begin{aligned}
C'_{[r+1]} &= C'_{[r]} + a_{2m+1} \\
&< r2^{m-1}RL + B + m(y_{2m} - y_1) \\
&= r2^{m\text{-}1}RL + \frac{1}{2}\sum_{k=1}^{2m} x_k + m\big[(m+1)R+3\big]2^{m\text{-}1}L + \sum_{j=1}^{m} 2^{j\text{-}1}L + m(x_{2m} - x_1) + (2^{m\text{-}1}-1)L \\
&< r2^{m-1}RL + 2L + m\big[(m+1)R + 3\big]2^{m-1}L + (2^m - 1)L + (2^{m-1} - 1)L \\
&= r2^{m-1}RL + (m+1)(mR+3)2^{m-1}L \\
&\leq r2^{m-1}RL + r(mR+3)2^{m-1}L \\
&= r\big[(m+1)R + 3\big]2^{m-1}L \\
&< C''_{[r]}.
\end{aligned}$$

Define $\Delta_1 = C''_{[r]} - C'_{[r+1]} > 0$. Let J_u denote the job immediately preceding J_{2m+1}. Interchanging J_u and J_{2m+1} in the schedule will increase the completion time of J_u by b_{2m+1} and will decrease the completion time of J_{2m+1} by min $\{b_u, b_{2m+1} + \Delta_1 + a_u\}$. Note that $b_u > b_{2m+1}$. Thus, interchanging J_u and J_{2m+1} can improve the total completion time of the schedule, which implies that the current schedule is not optimal.

Case 2: $r \leq m-1$. In this case, schedule σ must satisfy the following:

$$C'_{[r+1]} \geq a_{2m+1} = B + m(y_{2m} - y_1) > my_{2m} \geq (r+1)y_{2m} \geq \sum_{k=1}^{r} b_{[k]} + b_{[r+2]} = C''_{[r+2]} - b_{2m+1}.$$

Define $\Delta_2 = C''_{[r+2]} - C'_{[r+1]} < b_{2m+1}$. Let J_v denote the job immediately following J_{2m+1}. Interchanging J_{2m+1} and J_v in the schedule will decrease the completion time of J_v by at least b_{2m+1} and will increase the completion time of J_{2m+1} by at most $\max\{\Delta_2, a_v\}$. Note that $a_v < b_{2m+1}$. Thus, interchanging J_{2m+1} anc J_v can improve the total completion time of the schedule, which implies that the current schedule is not optimal. Combining Cases 1 and 2, we conclude that in schedule $\hat{\sigma}$, job J_{2m+1} must be scheduled in the $(m+1)$st position.

Lemma 3 implies that in schedule $\hat{\sigma}$, there are m jobs processed before the critical job and there are m jobs processed after it. Based on this result, we have the following lemma.

Lemma 4 *In schedule $\hat{\sigma}$, exactly one of J_{2j-1}, J_{2j} must be processed before J_{2m+1} for $j = 1, 2, \ldots, m$.*

Proof: Suppose, on the contrary, that schedule $\hat{\sigma}$ does not satisfy this property. Let

$$r = \max \Big\{ j \,\Big|\, \text{either both } J_{2j-1}, J_{2j} \text{ are processed before } J_{2m+1} \text{ or both } J_{2j-1}, J_{2j} \text{ are processed after } J_{2m+1} \Big\}.$$

Case 1: Both J_{2r-1}, J_{2r} precede J_{2m+1}. In this case, for $j = r+1, r+2, \ldots, m$, exactly one of J_{2j-1}, J_{2j} is processed before J_{2m+1}. These jobs contribute at least $(m-r)\left[(m+1)R+3\right]2^{m-1}L + \sum_{j=r+1}^{m} 2^{j-1} L = (m-r)\left[(m+1)R+3\right]2^{m-1}L +$

$(2^m - 2^r)L$ to the completion time of J_{2m+1} on M_2. The two jobs J_{2r-1}, J_{2r} contribute at least $2\,\big[\,(m+1)R+3\,\big]\;2^{m-1}\,L\,+\,2^rL$ to the completion time of $J_{2m\,+1}$ on M_2. In addition, there are $r-2$ other jobs processed before J_{2m+1}. These jobs contribute at least $(r-2)\big[(m+1)R+3\big]2^{m-1}L$ to the completion time of J_{2m+1} on M_2. Hence,

$$\begin{aligned}
C''_{[m+1]} &\geq (m-r)\big[(m+1)R+3\big]2^{m-1}L + (2^m - 2^r)L \\
&\quad + 2\big[(m+1)R+3\big]2^{m-1}L + 2^rL + (r-2)\big[(m+1)R+3\big]2^{m-1}L + b_{2m+1} \\
&= m\big[(m+1)R+3\big]2^{m-1}L + 2^mL + b_{2m+1} \\
&> \frac{1}{2}\sum_{k=1}^{2m} x_k + m\big[(m+1)R+3\big]2^{m-1}L + (2^m-1)L + b_{2m+1} \\
&= B + b_{2m+1}. \qquad (3)
\end{aligned}$$

Now, consider the total flow time of schedule $\hat{\sigma}$. The completion time of the critical job is at least $C''_{[m+1]}$. For $j = 1,\ 2,\ \ldots,\ m$, the completion time of the jth job is at least $\sum_{k=1}^{j} b_{[k]}$ and the completion time of the $(j+m+1)$ st job is at least $C''_{[m+1]} + \sum_{k=1}^{j} b_{[k+m+1]}$. Thus,

$$\begin{aligned}
Z(\hat{\sigma}) &\geq C''_{[m+1]} + \sum_{j=1}^{m}\sum_{k=1}^{j} b_{[k]} + \sum_{j=1}^{m}\left[C''_{[m+1]} + \sum_{k=1}^{j} b_{[k+m+1]}\right] \\
&= (m+1)C''_{[m+1]} + \sum_{j=1}^{m}(m-j+1)(b_{[j]} + b_{[j+m+1]}) \\
&\geq (m+1)C''_{[m+1]} + \sum_{j=1}^{m}(m-j+1)(y_{2j-1} + y_{2j}) \ \ (\text{since } y_1 < y_2 < \cdots < y_{2m}) \\
&> (m+1)(B + b_{2m+1}) + \sum_{j=1}^{m}(m-j+1)(y_{2j-1}+y_{2j}) \qquad (\text{by } (3)) \\
&= T,
\end{aligned}$$

which is a contradiction.

Case 2: J_{2m+1} precedes both J_{2r-1}, J_{2r}. In this case, for $j = r+1, r+2, \ldots, m$, exactly one of J_{2j-1}, J_{2j} is processed before J_{2m+1}. These jobs contribute at least $\sum_{j=r+1}^{m}(2^{m-1} - 2^{j-1})RL$ to the completion time of J_{2m+1} on M_1. In addition, r jobs from the subset $\{J_1,\ J_2, \ldots,\ J_{2r-2}\}$ are processed before J_{2m+1}. These jobs contribute at least $r2^{m-1}RL - 2\sum_{j=1}^{r-1} 2^{j-1}RL$ to the completion time of J_{2m+1} on M_1.This is because each a_k $(k=1,2,\ldots,2r)$ is at least $\lambda-\mu_k$, where $\lambda=2^{m-1}RL$ and $\mu_{2j-1}=\mu_{2j}=2^{j-1}$ RL $(j=1,2,\ldots,r-1)$. Therefore, the sum of the processing times of the a-tasks of any r jobs among $J_1,J_2,\ldots,J_{2r-2}$ is at least $r\lambda-\sum_{k=1}^{2r}\mu_k=r\lambda-\sum_{j=1}^{r}(\mu_{2j-1}+\mu_{2j})$ Hence,

$$
\begin{aligned}
C'_{[m+1]} &\geq \sum_{j=r+1}^{m}(2^{m-1}-2^{j-1})RL+\left[r2^{m-1}RL-2\sum_{j=1}^{r-1}2^{j-1}RL\right]+a_{2m+1}\\
&=\left[(m-r-2)2^{m-1}+2^r\right]RL+(r2^{m-1}-2^r+2)RL+a_{2m+1}\\
&=\left[(m-2)2^{m-1}+2\right]RL+B+m(y_{2m}-y_1). \qquad (4)
\end{aligned}
$$

Now, consider the total flow time of schedule $\hat{\sigma}$. The completion time of the critical job is at least $C'_{[m+1]}$. For $j=1,2,\ldots,m$, the completion time of the jth job is at least $\sum_{k=1}^{j}b_{[k]}$ and the completion time of the $(j+m+1)$ st job is at least $C''_{[m+1]}+\sum_{k=1}^{j}b_{[k+m+1]}$. Thus,

$$
\begin{aligned}
Z(\hat{\sigma}) &\geq C'_{[m+1]}+\sum_{j=1}^{m}\sum_{k=1}^{j}b_{[k]}+\sum_{j=1}^{m}\left[C''_{[m+1]}+\sum_{k=1}^{j}b_{[k+m+1]}\right]\\
&=C'_{[m+1]}+mC''_{[m+1]}+\sum_{j=1}^{m}(m-j+1)(b_{[j]}+b_{[j+m+1]})\\
&\geq C'_{[m+1]}+mC''_{[m+1]}+\sum_{j=1}^{m}(m-j+1)(y_{2j-1}+y_{2j}) \text{ (since } y_1<y_2<\cdots<y_{2m})\\
&=C'_{[m+1]}+m\left[\sum_{k=1}^{m}b_{[k]}+b_{2m+1}\right]+\sum_{j=1}^{m}(m-j+1)(y_{2j-1}+y_{2j})\\
&\geq C'_{[m+1]}+m^2\left[(m+1)R+3\right]2^{m-1}L+mb_{2m+1}+\sum_{j=1}^{m}(m-j+1)(y_{2j-1}+y_{2j})\\
&\geq \left[(m-2)2^{m-1}+2\right]RL+B+m(y_{2m}-y_1)\\
&\quad+m^2\left[(m+1)R+3\right]2^{m-1}L+mb_{2m+1}+\sum_{j=1}^{m}(m-j+1)(y_{2j-1}+y_{2j}) \text{ (by (4))}\\
&=T+RL+\left[(m-2)2^{m-1}+1\right]RL-mB\\
&\quad+m(y_{2m}-y_1)+m^2\left[(m+1)R+3\right]2^{m-1}L-b_{2m+1} \qquad \text{(by (1))}
\end{aligned}
$$

$$= T + RL - mB + m^2\big[(m+1)R + 3\big]2^{m-1}L - (m+1)K \text{ (by definition of } b_{2m+1})$$

$$= T + RL - \left\{\frac{m}{2}\sum_{k=1}^{2m} y_k - m^2\big[(m+1)R+3\big]2^{m-1}L + (m+1)K\right\} \text{ (by definition of } B)$$

$$= T + RL - \left[\frac{m}{2}\sum_{k=1}^{2m} x_k + m(2^m - 1)L + (m+1)K\right]$$

$$> T,$$

which is a contradiction. Therefore, for $j = 1, 2, \ldots, m$, exactly one of J_{2j-1}, J_{2j} must be processed before J_{2m+1}.

Next, we define $A_1 = \{k \mid J_k \text{ is processed before } J_{2m+1} \text{ in } \hat{\sigma}\}$ and $A_2 = A \setminus A_1$. Thus, by Lemma 4, $\{A_1, A_2\}$ is an even-odd partition of A. The next lemma implies that $\{A_1, A_2\}$ is the desired even-odd partition.

Lemma 5 $\sum_{k\in A_1} y_k = \sum_{k\in A_2} y_k = B$.

Proof: Since $Z(\hat{\sigma}) \leq T$, we have

$$\max\{C'_{[m+1]}, C''_{[m+1]}\} + mC''_{[m+1]} + \sum_{j=1}^{m}(m-j+1)(b_{[j]} + b_{[j+m+1]}) \leq T.$$

This implies that

$$(m+1)C''_{[m+1]} + \sum_{j=1}^{m}(m-j+1)(y_{2j-1} + y_{2j}) \leq T \tag{5}$$

and

$$C'_{[m+1]} + mC''_{[m+1]} + \sum_{j=1}^{m}(m-j+1)(y_{2j-1} + y_{2j}) \leq T. \tag{6}$$

Equation (5) implies that $C''_{[m+1]} \leq B + b_{2m+1}$, which in turn implies that $C''_{[m]} \leq B$, or equivalently,

$$\sum_{k\in A_1} y_k \leq B. \tag{7}$$

On the other hand, equation (6) implies that $C'_{[m+1]} + mC''_{[m+1]} \leq (m+1)(B + b_{2m+1})$, which in turn implies that

$$C'_{[m]} + mC''_{[m]} \leq (m+1)(B + b_{2m+1}) - a_{2m+1} - mb_{2m+1}.$$

Substituting the expressions for B, a_{2m+1}, and b_{2m+1} into this inequality and simplifying, we have

$$C'_{[m]} + mC''_{[m]} \leq \frac{m}{2}\sum_{k=1}^{2m} y_k + \big[(m-2)2^{m-1}+1\big]RL + (m+1)K. \quad (8)$$

Note that $A_1 = \{k \mid J_k \text{ is processed before } J_{2m+1} \text{ in } \hat{\sigma}\}$, which contains m elements (by Lemma 3). Thus, inequality (8) becomes

$$\sum_{k\in A_1} a_k + m\sum_{k\in A_1} b_k \leq \frac{m}{2}\sum_{k=1}^{2m} y_k + \big[(m-2)2^{m-1}+1\big]RL + (m+1)K. \quad (9)$$

Note also that

$$\begin{aligned}\sum_{k\in A_1} a_k &= \sum_{j=1}^{m}(2^{m-1}-2^{j-1})RL + \sum_{\substack{j=1,2,\ldots,m\\ \text{s.t. } 2j-1\in A_1}}(m+1)(y_{2j}-y_{2j-1})\\ &= \big[(m-2)2^{m-1}+1\big]RL + \sum_{\substack{j=1,2,\ldots,m\\ \text{s.t. } 2j-1\in A_1}}(m+1)(y_{2j}-y_{2j-1}).\end{aligned}$$

Substituting this into equation (9) and simplifying, we have

$$\sum_{\substack{j=1,2,\ldots,m\\ \text{s.t. } 2j-1\in A_1}}(m+1)(y_{2j}-y_{2j-1}) + m\sum_{k\in A_1} b_k \leq \frac{m}{2}\sum_{k=1}^{2m} y_k + (m+1)K,$$

or equivalently,

$$\sum_{\substack{j=1,2,\ldots,m\\ \text{s.t. } 2j-1\in A_1}}(m+1)(y_{2j}-y_{2j-1}) + m\sum_{k\in A_1} y_k \leq \frac{m}{2}\sum_{k=1}^{2m} y_k + \frac{m+1}{2}\sum_{j=1}^{m}(y_{2j}-y_{2j-1}).$$

Hence,

$$\frac{m+1}{2}\sum_{\substack{j=1,2,\ldots,m\\ \text{s.t. } 2j-1\in A_1}}(y_{2j}-y_{2j-1}) + \frac{m}{2}\sum_{k\in A_1} y_k \leq \frac{m}{2}\sum_{k\in A_2} y_k + \frac{m+1}{2}\sum_{\substack{j=1,2,\ldots,m\\ \text{s.t. } 2j-1\in A_2}}(y_{2j}-y_{2j-1}),$$

which implies that

$$\begin{aligned}&\frac{m+1}{2}\sum_{\substack{j=1,2,\ldots,m\\ \text{s.t. } 2j-1\in A_1}}(y_{2j}-y_{2j-1}) + \frac{m}{2}\left(\sum_{\substack{j=1,2,\ldots,m\\ \text{s.t. } 2j-1\in A_1}} y_{2j-1} + \sum_{\substack{j=1,2,\ldots,m\\ \text{s.t. } 2j-1\in A_2}} y_{2j}\right)\\ &\quad\leq \frac{m}{2}\left(\sum_{\substack{j=1,2,\ldots,m\\ \text{s.t. } 2j-1\in A_1}} y_{2j} + \sum_{\substack{j=1,2,\ldots,m\\ \text{s.t. } 2j-1\in A_2}} y_{2j-1}\right) + \frac{m+1}{2}\sum_{\substack{j=1,2,\ldots,m\\ \text{s.t. } 2j-1\in A_2}}(y_{2j}-y_{2j-1}).\end{aligned}$$

Upon simplification, we have $\frac{1}{2}\sum_{k\in A_2} y_k \leq \frac{1}{2}\sum_{k\in A_1} y_k$, or equivalently,

$$\sum_{k\in A_2} y_k \leq B. \tag{10}$$

Combining (7) and (10), we have $\sum_{k\in A_1} y_k = \sum_{k\in A_2} y_k = B$.

Summarizing the above analysis, we have the following result :**Theorem 1** *Problem* **P** *is NP-hard.*

3. Conclusion

We have proved that the two-machine scheduling problem with bundling operations and total flow time objective is NP-hard using a reduction from Even-Odd partition.

4. References

[1] Ahmadi, R., U. Bagchi and T.A. Roemer (2005). Coordinated scheduling of customer orders for quick response. Naval Research Logistics, **52,** 493–512.

[2] Cai, X. and X. Zhou (2004). Deterministic and stochastic scheduling with teamwork tasks. Naval Research Logistics, **51,** 818–840.

[3] Garey, M.R., R.E. Tarjan and G.T. Wilfong (1988). One-processor scheduling with symmetric earliness and tardiness penalties. Mathematics of Operations Research, **13,** 330–348.

[4] Lee, C.-Y., T.C.E. Cheng and B.M.T. Lin (1993). Minimizing the makespan in the 3-machine assembly-type flowshop scheduling problem. Management Science, **39,** 616–625.

[5] Leung, J.Y.-T., H. Li and M. Pinedo (2004). Order scheduling in an environment with dedicated resources in parallel. Journal of Scheduling, forthcoming.

[6] Leung, J.Y.-T., H. Li, M. Pinedo and C. Sriskandarajah (2005). Open shops with jobs overlap—revisited. European Journal of Operational Research, **163,** 569–571.

[7] Sung, C.S. and S.H. Yoon (1998). Minimizing total weighted completion time at a pre-assembly stage composed of two feeding machines. International Journal of Production Economics, **54,** 247–255.

[8] Wagneur, E. and C. Sriskandarajah (1993). Open shops with jobs overlap. European Journal of Operational Research, **71,** 366–378.

[9] Yang, J. (2005). The complexity of customer order scheduling problems on parallel machines. Computers and Operations Research, **32,** 1921–1939.