Journal of Scheduling (2019) 22:581–593
https://doi.org/10.1007/s10951-018-0598-5

# The complexity of CO-agent scheduling to minimize the total completion time and total number of tardy jobs

Rubing Chen[1] · Jinjiang Yuan[1] · Yuan Gao[1]

Published online: 17 December 2018

**Abstract**

In this paper, we revisit a two-agent scheduling problem on a single machine. In this problem, we have two competing agents $A$ and $B$, which means that the job set of agent $A$ and the job set of agent $B$ are disjoint. The objective is to minimize the total completion time of agent $A$, under the constraint that the total number of tardy jobs of agent $B$ is no larger than a given bound. The complexity of this problem was posed as open in Agnetis et al. (Oper Res 52:229–242, 2004). Leung et al. (Oper Res 58:458–469, 2010a, b. https://doi.org/10.1287/opre.1090.0744ec) showed that the problem is binary NP-hard. However, their NP-hardness proof has a flaw. Here, we present a new NP-hardness proof for this problem. Our research shows that the problem is still NP-hard even if the jobs of agent $A$ have a common processing time.

**Keywords** Two-agent scheduling · Total completion time · Total number of tardy jobs

## 1 Introduction

Suppose that there are two competing agents $A$ and $B$, which compete to perform their respective jobs on a common machine. For each $X \in \{A, B\}$, we use $\mathcal{J}^{(X)}$ to denote the set of jobs of agent $X$, and the jobs in $\mathcal{J}^{(X)}$ are called $X$-jobs. The assumption of "competing agents" means that $\mathcal{J}^{(A)} \cap \mathcal{J}^{(B)} = \emptyset$.

Suppose that $\mathcal{J}^{(A)} = \{J_1^{(A)}, J_2^{(A)}, \ldots, J_{n_A}^{(A)}\}$ and $\mathcal{J}^{(B)} = \{J_1^{(B)}, J_2^{(B)}, \ldots, J_{n_B}^{(B)}\}$. For $X \in \{A, B\}$, each $X$-job $J_j^{(X)} \in \mathcal{J}^{(X)}$ has a processing time $p_j^{(X)} > 0$. Each $B$-job $J_j^{(B)}$ also has a due date $d_j^{(B)}$. Given a feasible schedule $\sigma$, $C_j^{(X)}(\sigma)$ is the completion time of $J_j^{(X)}$, $X \in \{A, B\}$. $U_j^{(B)}(\sigma) = 1$ if $C_j^{(B)}(\sigma) > d_j^{(B)}$ and $U_j^{(B)}(\sigma) = 0$ if $C_j^{(B)}(\sigma) \le d_j^{(B)}$. For $X \in \{A, B\}$, let $f^{(X)}$ be the scheduling criterion of agent $X$ which depends only on the completion times of the $X$-jobs. Following the three-parameter notation introduced by Graham et al. (1979), the constrained scheduling problem on a single machine to minimize $f^{(A)}$ under the constraint that $f^{(B)}$ cannot exceed an upper bound $Q$ can be denoted by $1||f^{(A)} : f^{(B)} \le Q$.

The classical two-agent scheduling model was first introduced by Baker and Smith (2003) and Agnetis et al. (2004). Agnetis et al. (2004) considered various constrained scheduling problems for competing agents. They provided an $O(n \log n)$-time algorithm for problem $1||\sum C_j^{(A)} : f_{\max}^{(B)} \le Q$, an $O(n \log n)$-time algorithm for problem $1||\sum U_j^{(A)} : f_{\max}^{(B)} \le Q$, and an $O(n^3)$-time algorithm for problem $1||\sum U_j^{(A)} : \sum U_j^{(B)} \le Q$. But the computational complexity of problem $1||\sum C_j^{(A)} : \sum U_j^{(B)} \le Q$ was posed as open in Agnetis et al. (2004).

Ng et al. (2006) showed that problem $1||\sum C_j^{(A)} : \sum U_j^{(B)} \le Q$ is NP-hard under high-multiplicity (HM) encoding and can be solved in pseudo-polynomial time under binary encoding. HM encoding is an encoding system which was proposed in Hochbaum and Shamir (1991) and Clifford and Posner (2001). Under this system, the input length of $k$ identical jobs of processing time $p$ of the same type is just $O(\log(k+2) + \log(p+2))$. This means that the work in Ng et al. (2006), in fact, does not solve the open problem posed in Agnetis et al. (2004), since we usually study computational complexity under binary encoding.

Binary NP-hardness of problem $1||\sum C_j^{(A)} : \sum U_j^{(B)} \le Q$ was first presented in Leung et al. (2010), and a formal proof was published in Leung et al. (2010b). By using the NP-complete even–odd partition for the reduction, Leung et al. (2010b) showed that problem $1||\sum C_j^{(A)} : \sum U_j^{(B)} \le Q$ is

✉ Jinjiang Yuan
yuanjj@zzu.edu.cn

[1] School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001, Henan, People's Republic of China

**Table 1** Job data in instance $\mathcal{J}$

| Job | Processing time | Due date |
|---|---|---|
| $P_{2i-1}$ | $a_{2i-1}$ $(= p_{2i-1})$ | $\sum_{k=1}^{i-1} p_{2k} + \sum_{k=1}^{i-1} x_k + p_{2i-1}$ |
| $P_{2i}$ | $a_{2i} + (l_i - 1)\sigma_i$ $(= p_{2i})$ | $\sum_{k=1}^{i-1} p_{2k} + \sum_{k=1}^{i} x_k + p_{2i}$ |
| $R$ | $L$ | $\sum_{i=1}^{n} x_i + [H + \frac{1}{2}\sum_{i=1}^{n}(l_i - 1)\sigma_i] + L$ |
| $Q_i$ | $x_i$ | |

binary NP-hard. Unfortunately, their proof has a logical flaw. Thus we revisit problem $1||\sum C_j^{(A)} : \sum U_j^{(B)} \le Q$.

Although the NP-hardness proof in Leung et al. (2010b) is invalid, we happily find that the authors have in fact provided a reasonable and wonderful procedure for the NP-hardness proof, which is very useful in our research. In this paper, we first show that a special version of the even–odd partition is NP-complete. Then, by using an arbitrary instance of the special version of the even–odd partition for the reduction, we construct a new job instance for problem $1||\sum C_j^{(A)} : \sum U_j^{(B)} \le Q$. Finally, by borrowing the proof procedure provided in Leung et al. (2010b), we show that problem $1||\sum C_j^{(A)} : \sum U_j^{(B)} \le Q$ is NP-hard even if the jobs of agent $A$ have a common processing time.

In Sect. 2, by constructing a counterexample, we point out the logical flaw in the NP-hardness proof in Leung et al. (2010b). In Sect. 3, we present the NP-hardness proof of problem $1|p_j^{(A)} = p^{(A)}|\sum C_j^{(A)} : \sum U_j^{(B)} \le Q$.

## 2 The logical flaw and a counterexample

The following is the well-known even–odd partition problem. By Garey and Johnson (1979), the even–odd partition is binary NP-complete.

*Even–Odd Partition* Given a set of $2n + 1$ positive integers $a_1, a_2, \ldots, a_{2n}$ and $H$ such that $H = \frac{1}{2}\sum_{j=1}^{2n} a_j$, does there exist a partition $(I_1, I_2)$ of the index set $\{1, 2, \ldots, 2n\}$ such that $|I_1 \cap \{2j-1, 2j\}| = 1$ and $|I_2 \cap \{2j-1, 2j\}| = 1$ for each $j = 1, 2, \ldots, n$, and $\sum_{j\in I_1} a_j = \sum_{j\in I_2} a_j = H$?

By using the even–odd partition for the reduction, Leung et al. (2010b) showed that problem $1||\sum C_j^{(A)} : \sum U_j^{(B)} \le Q$ is binary NP-hard. Let us recall some related discussions in Leung et al. (2010b).

Let $\sigma_i = a_{2i} - a_{2i-1}$, $i = 1, 2, \ldots, n$. Since each pair of integers $\{a_{2i-1}, a_{2i}\}$ must be put into two different sets, Leung et al. (2010b) assumed that the given instance of the even–odd partition satisfies the following three properties:

**Property 1** $a_1 > (2n + 2)\max(\sigma_1, \ldots, \sigma_n)$.

**Property 2** $a_{2i-1} > \sum_{j=1}^{2i-2} a_j$.

**Property 3** $a_i/j$ is an integer for each $1 \le i \le 2n$ and $1 \le j \le n$.

For a given instance of the even–odd partition, Leung et al. (2010b) constructed an instance $\mathcal{J}$ of the scheduling problem $1||\sum C_j^{(A)} : \sum U_j^{(B)} \le Q$ with $3n + 1$ jobs: $2n$ $P$-jobs and a large $R$-job for agent $B$, and $n$ $Q$-jobs for agent $A$. The processing times and due dates for these jobs are shown in Table 1.

- $L$ is an integer larger than $2H$.
- $x_1 = 1$, $x_i = \frac{n-i+1}{n-i+2}a_{2i-3}$ for $i = 2, \ldots, n-1$, and $x_n = \frac{1}{2}a_{2n-3} + a_{2n-5}$. Note that $x_1 < x_2 < \cdots < x_n$, and they are all integers.
- $l_i\sigma_i = \frac{1}{n-i+1}a_{2i-1}$, $i = 1, 2, \ldots, n$.

Let the threshold for the total completion time of agent $A$ be $TC$, where

$$TC = \sum_{i=1}^{n}(n-i)[a_{2i} + (l_i - 1)\sigma_i] + \sum_{i=1}^{n}(n-i+1)x_i + \frac{1}{2}\sum_{i=1}^{n} l_i\sigma_i = H + \frac{1}{2}\sum_{i=1}^{n}(l_i - 1)\sigma_i,$$

and let the threshold for the number of tardy jobs of agent $B$ be $n$. The decision problem asks whether there is a feasible schedule for instance $\mathcal{J}$ such that $\sum C_j^{(A)} \le TC$ and $\sum U_j^{(B)} \le n$.

**Observation 2.1** *A partition $(I_1, I_2)$ of $\{1, 2, \ldots, 2n\}$ with $|I_1 \cap \{2i-1, 2i\}| = 1$ and $|I_2 \cap \{2i-1, 2i\}| = 1$ for each $i = 1, 2, \ldots, n$ is a solution for the instance of the even–odd partition if and only if the following condition (C1) holds.*

$$\sum_{2i-1\in I_1} \sigma_i = \sum_{2i-1\in I_2} \sigma_i = \frac{1}{2}\sum_{i=1}^{n} \sigma_i. \qquad (C1)$$

***Proof*** Let $(I_1, I_2)$ be a partition of $\{1, 2, \ldots, 2n\}$ such that $|I_1 \cap \{2i-1, 2i\}| = 1$ and $|I_2 \cap \{2i-1, 2i\}| = 1$ for each $i = 1, 2, \ldots, n$. Then, $(I_1, I_2)$ is a solution for the instance of the even–odd partition if and only if $\sum_{j\in I_1} a_j = \sum_{j\in I_2} a_j = H$, which is equivalent to the weakened version $\sum_{j\in I_1} a_j = \sum_{j\in I_2} a_j$, since $H = \frac{1}{2}\sum_{j=1}^{2n} a_j$. Now, we have

$$\begin{aligned}
&\sum_{j\in I_1} a_j = \sum_{j\in I_2} a_j\\
\Leftrightarrow &\sum_{i:2i-1\in I_1} a_{2i-1} + \sum_{i:2i\in I_1} a_{2i} = \sum_{i:2i-1\in I_2} a_{2i-1} + \sum_{i:2i\in I_2} a_{2i}\\
\Leftrightarrow &\sum_{i:2i-1\in I_1} a_{2i-1} + \sum_{i:2i-1\in I_2} a_{2i} = \sum_{i:2i-1\in I_2} a_{2i-1} + \sum_{i:2i-1\in I_1} a_{2i}\\
\Leftrightarrow &\sum_{i:2i-1\in I_1} a_{2i-1} - \sum_{i:2i-1\in I_1} a_{2i} = \sum_{i:2i-1\in I_2} a_{2i-1} - \sum_{i:2i-1\in I_2} a_{2i}\\
\Leftrightarrow &\sum_{i:2i-1\in I_1} \sigma_i = \sum_{i:2i-1\in I_2} \sigma_i\\
\Leftrightarrow &\sum_{i:2i-1\in I_1} \sigma_i = \sum_{i:2i-1\in I_2} \sigma_i = \frac{1}{2}\sum_{i=1}^{n}\sigma_i,
\end{aligned}$$

as required in the observation. □

**Observation 2.2** Leung et al. (2010b) *presented the binary NP-hardness proof of problem* $1||\sum C_j^{(A)} : \sum U_j^{(B)} \le Q$ *by proving that the instance of the even–odd partition has a solution if and only if instance* $\mathcal{J}$ *has a feasible schedule* $\pi$ *such that* $\sum C_j^{(A)}(\pi) \le TC$ *and* $\sum U_j^{(B)}(\pi) \le n$. *If their Lemma 1 (pp. 3–4) and Lemma 10 (p. 15) are correct, then the following statement holds.*

**Statement 1** *For every partition* $(I_1, I_2)$ *of the index set* $\{1, 2, \ldots, 2n\}$ *with* $|I_1 \cap \{2i-1, 2i\}| = 1$ *and* $|I_2 \cap \{2i-1, 2i\}| = 1$ *for each* $i = 1, 2, \ldots, n$, *the above condition (C1) and the following condition (C2) are equivalent.*

$$\sum_{i:2i-1\in I_1} l_i\sigma_i = \sum_{i:2i-1\in I_2} l_i\sigma_i = \frac{1}{2}\sum_{i=1}^{n} l_i\sigma_i. \qquad (C2)$$

***Proof*** See "Appendix". □

However, Statement 1 is invalid. We illustrate this by the following instance $\mathcal{I}$ of the even–odd partition with $n = 4$.

The instance $\mathcal{I} = (a_1, a_2, \ldots, a_8, H)$ is obtained by setting $a_1 = 264$, $a_2 = 288$, $a_3 = 576$, $a_4 = 600$, $a_5 = 1848$, $a_6 = 1872$, $a_7 = 5544$, $a_8 = 5568$, and $H = 8280$.

We can easily observe that instance $\mathcal{I}$ satisfies the above three properties and has a solution $(I_1, I_2)$ with $I_1 = \{1, 3, 6, 8\}$ and $I_2 = \{2, 4, 5, 7\}$ such that $\sum_{j\in I_1} a_j = \sum_{j\in I_2} a_j = 8280 = H$. Recall that $\sigma_i = a_{2i} - a_{2i-1}$ and $l_i\sigma_i = \frac{1}{n-i+1}a_{2i-1}$ for each $i = 1, 2, \ldots, n$. We then have $\sigma_1 = \sigma_2 = \sigma_3 = \sigma_4 = 24$, $l_1\sigma_1 = \frac{1}{4}a_1 = 66$, $l_2\sigma_2 = \frac{1}{3}a_3 = 192$, $l_3\sigma_3 = \frac{1}{2}a_5 = 924$, and $l_4\sigma_4 = a_7 = 5544$. Thus

$$\sum_{2i-1\in I_1} \sigma_i = \sum_{2i-1\in I_2} \sigma_i = \frac{1}{2}\sum_{i=1}^{n}\sigma_i = 48,$$

$$\sum_{2i-1\in I_1} l_i\sigma_i = l_1\sigma_1 + l_2\sigma_2 = 66 + 192 = 258$$

and

$$\sum_{2i-1\in I_2} l_i\sigma_i = l_3\sigma_3 + l_4\sigma_4 = 924 + 5544 = 6468.$$

It follows that $\sum_{2i-1\in I_1} l_i\sigma_i \ne \sum_{2i-1\in I_2} l_i\sigma_i$. This means that Statement 1 is invalid.

The above discussion shows that the NP-hardness proof in Leung et al. (2010b) has a logical flaw (in their Lemmas 1 and 10). Moreover, on p. 5, lines 6–8, the authors wrote: "Thus, a feasible schedule with $\sum C_j^a \le TC$ is obtained when the total processing time of the on-time P-jobs is exactly $A + \frac{1}{2}\sum_{i=1}^{n}(l_i - 1)\sigma_i$. But this occurs only when there is a solution for the instance of the even–odd partition problem." By checking the context in Leung et al. (2010b), we find that the second sentence is confusing.

From Observations 2.1 and 2.2, we guess (or conclude) that the authors in Leung et al. (2010b) have mistakenly assumed that condition (C1) is equivalent to (C2). This may be the reason for the logical flaw in Leung et al. (2010b).

We cannot find a method to amend this flaw directly. The computational complexity of problem $1||\sum C_j^{(A)} : \sum U_j^{(B)} \le Q$ should then be restudied.

## 3 NP-hardness proof

The following is the well-known partition problem. By Garey and Johnson (1979), the partition is binary NP-complete.

*Partition* Given a set of $n$ positive integers $x_1, x_2, \ldots, x_n$ and a positive integer $E$ such that $E = \frac{1}{2}\sum_{j=1}^{n} x_j$, does there exist a partition $(I_1, I_2)$ of the index set $\{1, 2, \ldots, n\}$, such that $\sum_{j\in I_1} x_j = \sum_{j\in I_2} x_j = E$?

From the NP-completeness proof of the partition in Garey and Johnson (1979), the following result is observed.

**Lemma 3.1** *The partition is also binary NP-complete for the instances* $(x_1, x_2, \ldots, x_n, E)$ *in which the n positive integers* $x_1, x_2, \ldots, x_n$ *are mutually distinct.*

By using the partition for the reduction, we now show that a special version of the even–odd partition is also binary NP-complete.

**Lemma 3.2** *The even–odd partition is also binary NP-complete for the instances* $(a_1, a_2, \ldots, a_{2n}, H)$ *in which the n integers* $|a_2 - a_1|, |a_4 - a_3|, \ldots, |a_{2n} - a_{2n-1}|$ *are mutually distinct.*

***Proof*** Suppose that $\mathcal{I} = (x_1, x_2, \ldots, x_n, E)$ is an instance of the partition. From Lemma 3.1, we may assume that $x_1, x_2, \ldots, x_n$ are mutually distinct. We construct an instance $\mathcal{I}' = (a_1, a_2, \ldots, a_{2n}, H)$ of the even–odd partition in the following way:

$$a_{2i-1} = 1,\ a_{2i} = x_i + 1 \text{ for } i = 1, 2, \ldots, n,$$

and $H = E + n$.

Since $a_{2i} - a_{2i-1} = x_i$ for $i = 1, 2, \ldots, n$, instance $\mathcal{I}'$ guarantees that $|a_2 - a_1|, |a_4 - a_3|, \ldots, |a_{2n} - a_{2n-1}|$ are mutually distinct.

For each partition $(I_1, I_2)$ of the index set $\{1, 2, \ldots, n\}$, we define

$$I_1' = \{2i - 1 : i \in I_2,\ 1 \le i \le n\} \cup \{2i : i \in I_1,\ 1 \le i \le n\}$$

and

$$I_2' = \{2i - 1 : i \in I_1,\ 1 \le i \le n\} \cup \{2i : i \in I_2,\ 1 \le i \le n\}.$$

Then, $(I_1', I_2')$ is a partition of the index set $\{1, 2, \ldots, 2n\}$ such that $|I_1' \cap \{2j-1, 2j\}| = |I_2' \cap \{2j-1, 2j\}| = 1$ for each $j = 1, 2, \ldots, n$. Moreover, we have that $\sum_{j \in I_1'} a_j = \sum_{2i-1 \in I_1'} a_{2i-1} + \sum_{2i \in I_1'} a_{2i} = \sum_{i \in I_2} 1 + \sum_{i \in I_1}(x_i + 1) = \sum_{i \in I_1} x_i + |I_1| + |I_2| = \sum_{i \in I_1} x_i + n$. For $I_2'$, we have a similar result. We then have

$$\sum_{j \in I_1'} a_j = \sum_{i \in I_1} x_i + n \text{ and } \sum_{j \in I_2'} a_j = \sum_{i \in I_2} x_i + n.$$

First, suppose that $(I_1, I_2)$ is a solution for instance $\mathcal{I}$ of the partition. We then have that $\sum_{j \in I_1} x_j = \sum_{j \in I_2} x_j = E$. From the definition of $(I_1', I_2')$, $(I_1', I_2')$ is a partition of the index set $\{1, 2, \ldots, 2n\}$ such that $|I_1' \cap \{2j-1, 2j\}| = |I_2' \cap \{2j-1, 2j\}| = 1$ for each $j = 1, 2, \ldots, n$. Furthermore, we have that $\sum_{j \in I_1'} a_j = \sum_{i \in I_1} x_i + n = E + n = H$ and $\sum_{j \in I_2'} a_j = \sum_{i \in I_2} x_i + n = E + n = H$. That is, $\sum_{j \in I_1'} a_j = \sum_{j \in I_2'} a_j = H$. Hence, $(I_1', I_2')$ is a solution for instance $\mathcal{I}'$ of the even–odd partition.

Conversely, suppose that $(I_1', I_2')$ is a solution for instance $\mathcal{I}'$ of the even–odd partition. We then have that $(I_1', I_2')$ is a partition of the index set $\{1, 2, \ldots, 2n\}$ such that $\sum_{j \in I_1'} a_j = \sum_{j \in I_2'} a_j = H$ and $|I_1' \cap \{2j-1, 2j\}| = |I_2' \cap \{2j-1, 2j\}| = 1$ for each $j = 1, 2, \ldots, n$. From the definition of $(I_1', I_2')$, $(I_1, I_2)$ is a partition of the index set $\{1, 2, \ldots, n\}$. And we have that $\sum_{i \in I_1} x_i = \sum_{j \in I_1'} a_j - n = H - n = E$ and $\sum_{i \in I_2} x_i = \sum_{j \in I_2'} a_j - n = H - n = E$. That is, $\sum_{i \in I_1} x_i = \sum_{i \in I_2} x_i = E$. Hence, $(I_1, I_2)$ is a solution for instance $\mathcal{I}$ of partition.

From the above discussion, $(I_1, I_2)$ is a solution for instance $\mathcal{I}$ of the partition if and only if $(I_1', I_2')$ is a solution for instance $\mathcal{I}'$ of the even–odd partition. Consequently, this special version of the even–odd partition is binary NP-complete. The lemma follows. □

For the above special version of the even–odd partition in Lemma 3.2, we have the following observations:

- Deleting a pair $\{a_{2j-1}, a_{2j}\}$ with $a_{2j-1} = a_{2j}$ (if any) will result in an equivalent instance. Then we may assume that $a_{2j-1} \ne a_{2j}$.
- Exchanging (if necessary) the indices of $a_{2j-1}$ and $a_{2j}$ will not affect the problem. Then we may assume that $a_{2j-1} < a_{2j}$.
- Renumbering (if necessary) the $n$ pairs $\{a_{2j-1}, a_{2j}\}$, $j = 1, 2, \ldots, n$, will not affect the problem. From Lemma 3.2, we may assume that $a_2 - a_1 < a_4 - a_3 < \cdots < a_{2n} - a_{2n-1}$.
- Replacing (if necessary) each pair $\{a_{2j-1}, a_{2j}\}$ with a new pair $\{a_{2j-1} + K_j, a_{2j} + K_j\}$, $j = 1, 2, \ldots, n$, and replacing $H$ with $H + \sum_{j=1}^{n} K_j$ will result in a new equivalent instance, where $K_1, K_2, \ldots, K_n$ are $n$ nonnegative integers with polynomial size in that of $(a_1, a_2, \ldots, a_{2n}, H)$. By choosing suitable $K_1, K_2, \ldots, K_n$, we can guarantee that $a_1 + K_1 < a_2 + K_1 < a_3 + K_2 < a_4 + K_2 < \cdots < a_{2n-1} + K_n < a_{2n} + K_n$. Then we may assume that $a_1 < a_2 < \cdots < a_{2n}$. Note that, if necessary, we may also assume that $a_1$ is suitably large.
- Instance $(a_1, a_2, \ldots, a_{2n}, H)$ is equivalent to instance $(2a_1, 2a_2, \ldots, 2a_{2n}, 2H)$. Thus, we may assume that all the integers in instance $(a_1, a_2, \ldots, a_{2n}, H)$ are even.

Based on the above discussion, in the remainder of this paper we consider only the instance $(a_1, a_2, \ldots, a_{2n}, H)$ of the even–odd partition such that all the $2n + 1$ integers are even, and moreover,

$$n(a_{2n} - a_{2n-1}) < a_1 < a_2 < \cdots < a_{2n}, \quad (1)$$

and

$$2 \le a_2 - a_1 < a_4 - a_3 < \cdots < a_{2n} - a_{2n-1}. \quad (2)$$

We are ready to show the NP-hardness of problem $1|p_j^{(A)} = p^{(A)}|\sum C_j^{(A)} : \sum U_j^{(B)} \le Q$.

**Theorem 3.1** *Problem* $1|p_j^{(A)} = p^{(A)}|\sum C_j^{(A)} : \sum U_j^{(B)} \le Q$ *is binary NP-hard.*

***Proof*** For a given instance $\mathcal{I} = (a_1, a_2, \cdots, a_{2n}, H)$ of the even–odd partition with all the $2n+1$ integers in the instance being even and with the two properties in (1) and (2), we construct a job instance $\mathcal{J}$ as follows: there are $3n+1$ jobs of two types, $n$ $A$-jobs $J_1^{(A)}, J_2^{(A)}, \ldots, J_n^{(A)}$ and $2n+1$ $B$-jobs $J_1^{(B)}, J_2^{(B)}, \ldots, J_{2n}^{(B)}, J_R^{(B)}$, where each $J_j^{(B)}$ with $1 \le j \le 2n$ is called a normal $B$-job, and $J_R^{(B)}$ is called the restricted $B$-job. The processing times and due dates are displayed in Table 2.

- $M_j = 3H(a_{2j} - a_{2j-1}) + (n - j)(a_{2j} - a_{2j-1}) - a_{2j-1}$ for $i = 1, 2, \ldots, n$.

**Table 2** The scheduling instance $\mathcal{J}$

| Job | Processing time | Due date |
|---|---|---|
| $J_j^{(A)}$ | $p_j^{(A)} = p^{(A)} = 1 + \sum_{k=1}^{n-1}(a_{2k} - a_{2k-1})$ | |
| $J_{2j-1}^{(B)}$ | $p_{2j-1}^{(B)} = M_j + a_{2j-1}$ | $d_{2j-1}^{(B)} = \sum_{k=1}^{j-1} p_{2k}^{(B)} + (j-1)p^{(A)} + p_{2j-1}^{(B)}$ |
| $J_{2j}^{(B)}$ | $p_{2j}^{(B)} = M_j + a_{2j}$ | $d_{2j}^{(B)} = \sum_{k=1}^{j-1} p_{2k}^{(B)} + jp^{(A)} + p_{2j}^{(B)}$ |
| $J_R^{(B)}$ | $p_R^{(B)} = L$ | $d_R^{(B)} = \sum_{j=1}^{n} M_j + np^{(A)} + H + L$ |

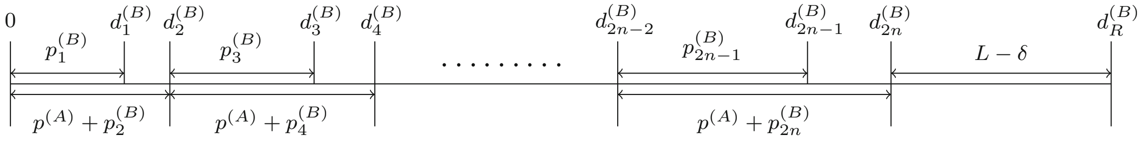

**Fig. 1** The due dates for $B$-jobs in instance $\mathcal{J}$

- $L = n^2 p^{(A)} + n \sum_{j=1}^{2n} p_j^{(B)}$ is a sufficiently large number.

Note that the definitions of processing times and due dates for normal $B$-jobs can be rewritten together as

$$\begin{cases} p_{j'}^{(B)} = M_j + a_{j'}, \\ d_{j'}^{(B)} = \sum_{k=1}^{j-1} p_{2k}^{(B)} + \lfloor j'/2 \rfloor p^{(A)} + p_{j'}^{(B)} \end{cases} \tag{3}$$

for every $j' \in \{2j-1,\ 2j\}$, $j = 1, 2, \ldots, n$.

Let the upper bound of $\sum U_j^{(B)}$ be defined by $Q = n$, and the threshold value of $\sum C_j^{(A)}$ be given by

$$Y = \frac{n(n+1)}{2} p^{(A)} + \sum_{j=1}^{n} (n-j) p_{2j}^{(B)} + 3H\delta, \tag{4}$$

where $\delta = \frac{1}{2} \sum_{j=1}^{n} (a_{2j} - a_{2j-1})$. It is not hard to verify that $L > Y$.

The decision asks whether there is a feasible schedule $\sigma$ on the above job instance such that

$$\sum C_j^{(A)}(\sigma) \le Y \text{ and } \sum U_j^{(B)}(\sigma) \le n. \tag{5}$$

Clearly, the above construction can be done in polynomial time.

Note that $p^{(A)} < a_1$ and $p_{2j-1}^{(B)} < p_{2j}^{(B)}$ for $j = 1, 2, \ldots, n$. We then have $d_{2j}^{(B)} - d_{2j-1}^{(B)} = p^{(A)} + (a_{2j} - a_{2j-1})$ for $j = 1, 2, \ldots, n$. Figure 1 shows the structure of the due dates for all $B$-jobs in instance $\mathcal{J}$.

For convenience, we call $\mathcal{I} = (a_1, a_2, \cdots, a_{2n}, H)$ the *partition instance* in the sequel. We set $\mathcal{J}^{(A)} = \{J_1^{(A)}, J_2^{(A)}, \ldots, J_n^{(A)}\}$ and $\mathcal{J}^{(B)} = \{J_1^{(B)}, J_2^{(B)}, \ldots, J_{2n}^{(B)}, J_R^{(B)}\}$. Since the $n$ jobs in $\mathcal{J}^{(A)}$ are indistinguishable, in the sequel we consider only the schedules $\sigma$ for instance $\mathcal{J}$, in which the $A$-jobs are scheduled in the order of their indices in $\sigma$, i.e.,

$$J_1^{(A)} \prec_\sigma J_2^{(A)} \prec_\sigma \cdots \prec_\sigma J_n^{(A)}. \tag{6}$$

For each schedule $\sigma$ of $\mathcal{J}$ (or of $\mathcal{J}^{(B)}$), we use $\mathcal{O}(\sigma)$ to denote the set of on-time $B$-jobs in $\sigma$.

*Outline of our proof* According to the principle of NP-hardness, we need to show that the partition instance $\mathcal{I} = (a_1, a_2, \cdots, a_{2n}, H)$ has a solution if and only if there is a feasible schedule $\sigma$ of the scheduling instance $\mathcal{J}$ such that the two inequalities in Eq. (5) are satisfied. To this end, the following work will be done in the sequel:

- Some useful inequalities are established in Eqs. (7)–(11).
- Two important inequalities related to the due dates for $B$-jobs are established in Lemma 3.3.
- Nice sequences are introduced, followed by the establishment of some useful inequalities related to nice sequences in Lemma 3.4.
- Some useful properties of the optimal schedules of problem $1||\sum U_j^{(B)}$ are established in Lemma 3.5. From Lemma 3.5, we see that $\sum U_j^{(B)}(\sigma) \le n$ in (5) is equivalent to $\sum U_j^{(B)}(\sigma) = n$.
- From the previous equations and from Lemmas 3.3–3.5, some useful properties of the efficient schedules (which are feasible schedules of problem $1||\sum C_j^{(A)} : \sum U_j^{(B)} = n$ with $\sum C_j^{(A)} \le Y$) are established in Lemma 3.6.
- The schedules corresponding to the nice sequences are introduced, followed by the establishment of an exact expression for the value $\sum C_j^{(A)}(\sigma)$ in Eq. (19) for every such schedule $\sigma$.
- Finally, by using Lemma 3.6 and Eq. (19), we can show that the partition instance $\mathcal{I} = (a_1, a_2, \ldots, a_{2n}, H)$ has a solution if and only if the job instance $\mathcal{J}$ has an efficient schedule. The result then follows.

Now let us begin our proof. From the definition of $M_j$, for $j = 1, 2, \ldots, n-1$, we have

$$\begin{aligned} & M_{j+1} - M_j \\ &\quad = 3H(a_{2j+2} - a_{2j+1}) + (n - j - 1) \\ &\qquad (a_{2j+2} - a_{2j+1}) - a_{2j+1} \\ &\qquad -3H(a_{2j} - a_{2j-1}) - (n-j)(a_{2j} - a_{2j-1}) + a_{2j-1} \\ &\quad > 3H(a_{2j+2} - a_{2j+1}) - a_{2j+1} \\ &\qquad - 3H(a_{2j} - a_{2j-1}) - n(a_{2j} - a_{2j-1}) \\ &\quad \geq 6H - n(a_{2j} - a_{2j-1}) - a_{2j+1} \quad \text{(From (2))} \\ &\quad > 6H - a_1 - a_{2n} \quad \text{(From (1) and (2))} \\ &\quad > 4H > 0, \quad (\text{Since } 2H = a_1 + \cdots + a_{2n}) \end{aligned}$$

that is,

$$M_{j+1} - M_j > 4H > 0. \tag{7}$$

Consequently, we have

$$M_1 < M_2 < \cdots < M_n. \tag{8}$$

From the definitions of the due dates and processing times for $B$-jobs, we further have

$$3H < p_1^{(B)} < p_2^{(B)} < \cdots < p_{2n}^{(B)} < p_R^{(B)}, \tag{9}$$

and

$$d_1^{(B)} < d_2^{(B)} < \cdots < d_{2n}^{(B)} < d_R^{(B)}. \tag{10}$$

From (1) and (2), we have that $np^{(A)} < n^2(a_{2n} - a_{2n-1}) < na_1 < \sum_{k=1}^n a_{2k-1} < H$, that is,

$$H > np^{(A)}. \tag{11}$$

The following lemma establishes two important inequalities related to the due dates for $B$-jobs.

**Lemma 3.3** (i) $d_{2j}^{(B)} < 3H + p_1^{(B)} + p_3^{(B)} + \cdots + p_{2j-1}^{(B)}$, *for* $j = 1, 2, \ldots, n$, *and*
(ii) $d_R^{(B)} < 3H + p_1^{(B)} + p_3^{(B)} + \cdots + p_{2n-1}^{(B)} + p_R^{(B)}$.

***Proof*** For $j = 1, 2, \ldots, n$, we have

$$\begin{aligned} & 3H + p_1^{(B)} + p_3^{(B)} + \cdots + p_{2j-1}^{(B)} \\ &\quad = 2H + H + \sum_{k=1}^{j} (M_k + a_{2k-1}) \\ &\quad > \sum_{k=1}^{j} (M_k + a_{2k-1}) + 2H + np^{(A)} \quad \text{(From (11))} \\ &\quad > \sum_{k=1}^{j} (M_k + a_{2k}) + np^{(A)} \\ &\quad > d_{2j}^{(B)}. \end{aligned}$$

This proves (i).

Note that we also have

$$\begin{aligned} & 3H + p_1^{(B)} + p_3^{(B)} + \cdots + p_{2n-1}^{(B)} + p_R^{(B)} \\ &\quad = H + 2H + \sum_{k=1}^{n} (M_k + a_{2k-1}) + L \\ &\quad > np^{(A)} + \sum_{k=1}^{n} M_k + 2H + L \quad \text{(From (11))} \\ &\quad > np^{(A)} + \sum_{k=1}^{n} M_k + H + L \\ &\quad = d_R^{(B)}. \end{aligned}$$

This proves (ii). The lemma follows. □

For an index $j \in \{1, 2, \ldots, n\}$, a sequence $(1', 2', \ldots, j')$ of indices in $\{1, 2, \ldots, 2j\}$ is called a *$j$-nice sequence* if $k' \in \{2k-1, 2k\}$ for $k = 1, 2, \ldots, j$. An $n$-nice sequence is also called a *nice sequence*. It is easy to see that if $(1', 2', \ldots, n')$ is a nice sequence, then $(1', 2', \ldots, j')$ is a $j$-nice sequence for $j = 1, 2, \ldots, n$. The following lemma establishes some useful inequalities related to nice sequences.

**Lemma 3.4** *Let* $(1', 2', \ldots, j')$ *be a $j$-nice sequence, where* $j \in \{1, 2, \ldots, n\}$. *Then we have the following inequalities:*

(i) $\lfloor j'/2 \rfloor p^{(A)} + \sum_{k=1}^{j} p_{k'}^{(B)} \leq d_{j'}^{(B)}$;
(ii) $np^{(A)} + \sum_{k=1}^{n} p_{k'}^{(B)} + p_R^{(B)} \leq d_R^{(B)}$ *if and only if* $\sum_{k=1}^{n} a_{k'} \leq H$;
(iii) $(j+1)p^{(A)} + \sum_{k=1}^{j} p_{k'}^{(B)} > d_{j'}^{(B)}$ *if* $j \leq n-1$.

***Proof*** The proof will follow from Eqs. (3), (9), and (10) and the definition of $d_R^{(B)}$.

The result in (i) holds, since $\lfloor j'/2 \rfloor p^{(A)} + \sum_{k=1}^{j} p_{k'}^{(B)} \leq \lfloor j'/2 \rfloor p^{(A)} + \sum_{k=1}^{j-1} p_{2k}^{(B)} + p_{j'}^{(B)} = d_{j'}^{(B)}$.

The result in (ii) follows by noting that $np^{(A)} + \sum_{k=1}^{n} p_{k'}^{(B)} + p_R^{(B)} = np^{(A)} + \sum_{k=1}^{n} M_k + L + \sum_{k=1}^{n} a_{k'}$ and $d_R^{(B)} = np^{(A)} + \sum_{k=1}^{n} M_k + L + H$.

The result in (iii) holds, since $(j+1)p^{(A)} + \sum_{k=1}^{j} p_{k'}^{(B)} \geq jp^{(A)} + 1 + \sum_{k=1}^{n-1} (a_{2k} - a_{2k-1}) + \sum_{k=1}^{j} (M_k + a_{2k-1}) \geq jp^{(A)} + \sum_{k=1}^{j} (M_k + a_{2k}) + 1 = d_{2j}^{(B)} + 1 > d_{2j}^{(B)}$. The lemma follows. □

To proceed with our proof, we next establish two lemmas which will be used in our discussion.

**Lemma 3.5** *For problem $1||\sum U_j^{(B)}$ on B-jobs $\mathcal{J}^{(B)}$ (without considering A-jobs), every optimal schedule $\pi$ has the following properties:*

(i) *the (optimal) value of $\pi$ is given by $\sum U_j^{(B)}(\pi) = n$;*

(ii) *exactly one job of each pair $\{J_{2j-1}^{(B)}, J_{2j}^{(B)}\}$, $1 \le j \le n$, is on time in $\pi$;*

(iii) *$J_R^{(B)}$ is on time in $\pi$ and is scheduled after all the other on-time B-jobs in $\pi$;*

(iv) *all on-time B-jobs are scheduled in the EDD order in $\pi$;*

(v) *all tardy B-jobs are scheduled after the restricted B-job $J_R^{(B)}$ in $\pi$.*

***Proof*** Property (i) can be proved by applying the algorithm presented by Moore (1968) for the problem $1||\sum U_j$. In what follows, we will prove it in a different way to save space.

We first prove the following weakened version of property (i):

(i′) *the optimal value of problem $1||\sum U_j^{(B)}$ on the instance of B-jobs is at most n.*

To prove (i′), we consider the schedule $\sigma = (J_1^{(B)}, J_3^{(B)}, \ldots, J_{2n-1}^{(B)}, J_R^{(B)}, J_2^{(B)}, J_4^{(B)}, \ldots, J_{2n}^{(B)})$. Since $J_R^{(B)}$ has a very large processing time $L$, the $n$ B-jobs $J_2^{(B)}, J_4^{(B)}, \ldots, J_{2n}^{(B)}$ are tardy in $\sigma$. We will show that the other $n+1$ B-jobs $J_1^{(B)}, J_3^{(B)}, \ldots, J_{2n-1}^{(B)}, J_R^{(B)}$ are on time in $\sigma$.

From Lemma 3.4(i), by setting $j' = 2j-1$ for each $j$ with $1 \le j \le n$, we have $C_{2j-1}^{(B)}(\sigma) = \sum_{k=1}^{j} p_{2k-1}^{(B)} \le d_{2j-1}^{(B)}$. This implies that $J_1^{(B)}, J_3^{(B)}, \ldots, J_{2n-1}^{(B)}$ are on time in $\sigma$.

Note that $C_R^{(B)}(\sigma) = \sum_{k=1}^{n} p_{2k-1}^{(B)} + p_R^{(B)} < np^{(A)} + \sum_{k=1}^{n} p_{2k-1}^{(B)} + p_R^{(B)}$. Since $\sum_{k=1}^{n} a_{2k-1} < H$, from Lemma 3.4(ii), we have $C_R^{(B)}(\sigma) < d_R^{(B)}$. Thus, $J_R^{(B)}$ is on time in $\sigma$. This proves property (i′).

Now we consider an optimal schedule $\pi$ for problem $1||\sum U_j^{(B)}$ on B-jobs $\mathcal{J}^{(B)}$. Let $\mathcal{O}(\pi)$ be the set of on-time B-jobs in $\pi$.

By contradiction, suppose that property (ii) does not hold for $\pi$. Let $j$ be the smallest index in $\{1, 2, \ldots, n\}$ so that $\{J_{2j-1}^{(B)}, J_{2j}^{(B)}\}$ violates property (ii). Then, for each $k$ with $1 \le k \le j-1$, exactly one job of each pair $\{J_{2k-1}^{(B)}, J_{2k}^{(B)}\}$ is on time in $\pi$, and moreover, $J_{2j-1}^{(B)}$ and $J_{2j}^{(B)}$ are either both tardy or both on time in $\pi$.

From property (i′), there are at least $n$ on-time jobs among $\{J_1^{(B)}, J_2^{(B)}, \ldots, J_{2n}^{(B)}\}$ in $\pi$, or equivalently, $|\mathcal{O}(\pi) \cap \{J_1^{(B)}, J_2^{(B)}, \ldots, J_{2n}^{(B)}\}| \ge n$. Let $v$ be the smallest index in $\{j, j+1, \ldots, n\}$ such that $|\mathcal{O}(\pi) \cap \{J_1^{(B)}, J_2^{(B)}, \ldots, J_{2v}^{(B)}\}| \ge v$.

If $v = j$, by the choices of $j$ and $v$, $|\mathcal{O}(\pi) \cap \{J_{2k-1}^{(B)}, J_{2k}^{(B)}\}| = 1$ for $k = 1, 2, \ldots, j-1$ and $\{J_{2j-1}^{(B)}, J_{2j}^{(B)}\} \subseteq \mathcal{O}(\pi)$. Thus, from (9), (10), and Lemma 3.3(i), we have

$$\begin{aligned} d_{2j}^{(B)} &= \max\{d_i^{(B)} : i = 1, 2, \ldots, 2j\} \\ &\ge \max\{C_i^{(B)}(\pi) : J_i^{(B)} \in \mathcal{O}(\pi) \cap \{J_1^{(B)}, J_2^{(B)}, \ldots, J_{2j}^{(B)}\}\} \\ &\ge \sum_{k=1}^{j-1} \min\{p_{2k-1}^{(B)}, p_{2k}^{(B)}\} + p_{2j-1}^{(B)} + p_{2j}^{(B)} \\ &= \sum_{k=1}^{j-1} p_{2k-1}^{(B)} + p_{2j-1}^{(B)} + p_{2j}^{(B)} \\ &> 3H + \sum_{k=1}^{j} p_{2k-1}^{(B)} \\ &> d_{2j}^{(B)}, \end{aligned}$$

a contradiction. Consequently, we have

$$v > j \text{ and } \mathcal{O}(\pi) \cap \{J_{2j-1}^{(B)}, J_{2j}^{(B)}\} = \emptyset. \tag{12}$$

From the definition of $v$, we further have

$$|\mathcal{O}(\pi) \cap \{J_1^{(B)}, J_2^{(B)}, \ldots, J_{2v}^{(B)}\}| = v. \tag{13}$$

Now we suppose that $\mathcal{O}(\pi) \cap \{J_1^{(B)}, J_2^{(B)}, \ldots, J_{2v}^{(B)}\} = \{J_{1'}^{(B)}, J_{2'}^{(B)}, \ldots, J_{v'}^{(B)}\}$ so that $1' < 2' < \cdots < v'$. From the choices of $j$ and $v$, together with (12) and (13), we have

$$\begin{cases} J_{k'}^{(B)} \in \{J_{2k-1}^{(B)}, J_{2k}^{(B)}\}, & \text{for } k = 1, 2, \ldots, j-1, \\ J_{k'}^{(B)} \in \{J_{2k+1}^{(B)}, J_{2k+2}^{(B)}, \ldots, J_{2v-2}^{(B)}\}, & \text{for } k = j, j+1, \ldots, v-2, \\ J_{(v-1)'}^{(B)} = J_{2v-1}^{(B)}, \\ J_{v'}^{(B)} = J_{2v}^{(B)}. \end{cases}$$

Consequently, from the definitions of $J_{2j-1}^{(B)}$ and $J_{2j}^{(B)}$ and (9), we have

$$\begin{cases} p_{k'}^{(B)} \ge p_{2k-1}^{(B)} = M_k + a_{2k-1}, & \text{for } k = 1, 2, \ldots, j-1, \\ p_{k'}^{(B)} \ge p_{2k+1}^{(B)} = M_{k+1} + a_{2k+1}, & \text{for } k = j, j+1, \ldots, v-2, \\ p_{(v-1)'}^{(B)} = p_{2v-1}^{(B)} = M_v + a_{2v-1}, \\ p_{v'}^{(B)} = p_{2v}^{(B)} = M_v + a_{2v}. \end{cases} \tag{14}$$

Note that $v \ge j+1$. From (7), we have $p_{2v}^{(B)} = M_v + a_{2v} > M_j + a_{2j-1} + 4H = p_{2j-1}^{(B)} + 4H > p_{2j-1}^{(B)} + 3H$. From (9), (10), (14), and Lemma 3.3(i), we have

$$\begin{aligned} d_{2v}^{(B)} &= \max\{d_{k'}^{(B)} : k = 1, 2, \ldots, v\} \\ &\ge \max\{C_{k'}^{(B)}(\pi) : k = 1, 2, \ldots, v\} \\ &\ge \sum_{k=1}^{v} p_{k'}^{(B)} \end{aligned}$$

$$\begin{aligned}
&= \sum_{k=1}^{j-1} p_{k'}^{(B)} + \sum_{k=j}^{v-2} p_{k'}^{(B)} + p_{(v-1)'}^{(B)} + p_{v'}^{(B)} \\
&\geq \sum_{k=1}^{j-1} p_{2k-1}^{(B)} + \sum_{k=j+1}^{v-1} p_{2k-1}^{(B)} + p_{2v-1}^{(B)} + p_{2v}^{(B)} \\
&> 3H + \sum_{k=1}^{v} p_{2k-1}^{(B)} \\
&> d_{2v}^{(B)},
\end{aligned}$$

a contradiction. Property (ii) follows.

From properties (i′) and (ii), the restricted $B$-job $J_R^{(B)}$ must be on time in $\pi$. Since $p_R^{(B)} = L$ is a sufficiently large integer, all the other on-time $B$-jobs must be scheduled before $J_R^{(B)}$. This proves property (iii).

From properties (i′), (ii), and (iii), we certainly have $\sum U_j^{(B)}(\pi) = n$, and so property (i) follows.

The above three properties enable us to define $J_{j'}^{(B)}$, $1 \leq j \leq n$, to be the unique job in $\mathcal{O}(\pi) \cap \{J_{2j-1}^{(B)}, J_{2j}^{(B)}\}$. Note that, from property (iii), $J_R^{(B)}$ is the last on-time job in $\pi$.

To prove property (iv), by contradiction, we suppose that there are two indices $i$ and $j$ with $d_{i'}^{(B)} < d_{j'}^{(B)}$ such that $J_{i'}^{(B)}$ is scheduled after $J_{j'}^{(B)}$ in $\pi$. For our purpose, we may choose such a pair $(i, j)$ so that $i$ is as small as possible. From (10), we have $1 \leq i < j \leq n$. By the choice of $(i, j)$, all the jobs $J_{k'}^{(B)}$, $1 \leq k \leq i$, and $J_{j'}^{(B)}$ are completed by time $C_{i'}^{(B)}(\pi)$ in $\pi$. We then have

$$\begin{aligned}
C_{i'}^{(B)}(\pi) &\geq \sum_{k=1}^{i} p_{k'}^{(B)} + p_{j'}^{(B)} \\
&\geq 3H + \sum_{k=1}^{i} p_{2k-1}^{(B)} \quad \text{(From (9))} \\
&> d_{2i}^{(B)} \quad \text{(From Lemma 3.3(i))} \\
&\geq d_{i'}^{(B)}. \quad \text{(From (10))}
\end{aligned}$$

This contradicts our assumption that $J_{i'}^{(B)}$ is on time in $\pi$. Property (iv) follows.

Finally, if property (v) is violated, there must be a tardy $B$-job $J_x^{(B)}$ which is scheduled before the restricted $B$-job $J_R^{(B)}$ in $\pi$. Then all the jobs $J_{k'}^{(B)}$, $1 \leq k \leq n$, $J_x^{(B)}$ and $J_R^{(B)}$ are completed by time $C_R^{(B)}(\pi)$ in $\pi$. As a result, we have

$$\begin{aligned}
C_R^{(B)}(\pi) &\geq \sum_{k=1}^{n} p_{k'}^{(B)} + p_R^{(B)} + p_x^{(B)} \\
&\geq 3H + \sum_{k=1}^{n} p_{2k-1}^{(B)} + p_R^{(B)} \quad \text{(From (9))} \\
&> d_R^{(B)}. \quad \text{(From Lemma 3.3(ii))}
\end{aligned}$$

This contradicts property (iii). The lemma follows. □

From Lemma 3.5, $\sum U_j^{(B)}(\sigma) \leq n$ in (5) is equivalent to $\sum U_j^{(B)}(\sigma) = n$. Thus, we define an *efficient schedule* to be a feasible schedule $\sigma$ for instance $\mathcal{J}$ such that

$$\sum C_j^{(A)}(\sigma) \leq Y \text{ and } \sum U_j^{(B)}(\sigma) = n. \tag{15}$$

**Lemma 3.6** *Suppose that instance $\mathcal{J}$ has efficient schedules, and let $\sigma$ be an efficient schedule for instance $\mathcal{J}$ such that $\sum C_j^{(A)}(\sigma)$ is as small as possible. Then $\sigma$ has the following properties:*

(i) *the properties (ii)–(v) in Lemma 3.5 are still valid for $\sigma$;*

(ii) *all the $A$-jobs are scheduled before $J_R^{(B)}$ in $\sigma$;*

(iii) *let $J_{j'}^{(B)}$, $1 \leq j \leq n$, be the on-time $B$-job in $\{J_{2j-1}^{(B)}, J_{2j}^{(B)}\}$. Then $J_{j'}^{(B)} \prec_\sigma J_{j+1}^{(A)}$ and $J_j^{(A)} \prec_\sigma J_{(j+1)'}^{(B)}$ for $j = 1, 2, \ldots, n-1$;*

(iv) *for each $j = 1, 2, \ldots, n$, there are only two possible configurations for the triple $\{J_{2j-1}^{(B)}, J_j^{(A)}, J_{2j}^{(B)}\}$ in $\sigma$:*
   - *either $J_{2j-1}^{(B)}$ is on time, $J_{2j}^{(B)}$ is tardy, and $J_{2j-1}^{(B)}$ and $J_j^{(A)}$ are scheduled consecutively in this order,*
   - *or $J_{2j-1}^{(B)}$ is tardy, $J_{2j}^{(B)}$ is on time, and $J_j^{(A)}$ and $J_{2j}^{(B)}$ are scheduled consecutively in this order.*

***Proof*** Suppose that $\sigma$ is an efficient schedule for instance $\mathcal{J}$ such that $\sum C_j^{(A)}(\sigma)$ is as small as possible. Then $\sum U_j^{(B)}(\sigma) = n$. This means that the schedule obtained from $\sigma$ by deleting the $A$-jobs is also an optimal schedule for problem $1||\sum U_j^{(B)}$ on $B$-jobs $\mathcal{J}^{(B)}$. Consequently, the properties (ii)–(v) in Lemma 3.5 are still valid for $\sigma$.

Since $p_R^{(B)} = L > Y$ is a sufficiently large integer, all $A$-jobs must be scheduled before $J_R^{(B)}$ in $\sigma$. Property (ii) follows.

Recall that $\mathcal{O}(\sigma)$ is the set of on-time $B$-jobs in $\sigma$. The above discussion enables us to define $J_{j'}^{(B)}$, $1 \leq j \leq n$, to be the unique job in $\mathcal{O}(\sigma) \cap \{J_{2j-1}^{(B)}, J_{2j}^{(B)}\}$. Then $(1', 2', \cdots, n')$ is a nice sequence. From property (i) and Lemma 3.5, in schedule $\sigma$, the on-time $B$-jobs are scheduled in the order

$$J_{1'}^{(B)} \prec_\sigma J_{2'}^{(B)} \prec_\sigma \cdots \prec_\sigma J_{n'}^{(B)}. \tag{16}$$

Recall that $A$-jobs are scheduled in the order as (6), i.e.,

$$J_1^{(A)} \prec_\sigma J_2^{(A)} \prec_\sigma \cdots \prec_\sigma J_n^{(A)}.$$

To prove property (iii), by contradiction, we suppose that there is some $j \in \{1, 2, \ldots, n-1\}$ so that either $J_{j+1}^{(A)} \prec_\sigma J_{j'}^{(B)}$ or $J_{(j+1)'}^{(B)} \prec_\sigma J_j^{(A)}$.

If $J^{(A)}_{j+1} \prec_\sigma J^{(B)}_{j'}$, from (6) and (16), we have $\{J^{(A)}_1, \cdots, J^{(A)}_{j+1}\} \prec_\sigma J^{(B)}_{j'}$, and $\{J^{(B)}_{1'}, \cdots, J^{(B)}_{(j-1)'}\} \prec_\sigma J^{(B)}_{j'}$. Thus, from Lemma 3.4(iii), we have $C^{(B)}_{j'}(\sigma) \geq (j+1)p^{(A)} + \sum_{k=1}^{j} p^{(B)}_{k'} > d^{(B)}_{j'}$. This contradicts our assumption that $J^{(B)}_{j'}$ is on time in $\sigma$.

Now, by the choice of $j$, we must have $J^{(B)}_{(j+1)'} \prec_\sigma J^{(A)}_j$. From (6) and (16), again, there must be a pair of indices $x$ and $y$ with $1 \leq x < y \leq n$ such that $J^{(B)}_{y'} \prec_\sigma J^{(A)}_x$ and $J^{(B)}_{y'}$ and $J^{(A)}_x$ are consecutively scheduled in $\sigma$, i.e., $C^{(B)}_{y'}(\sigma) = S^{(A)}_x(\sigma)$, where $S^{(A)}_x(\sigma)$ is the starting time of $J^{(A)}_x$ in $\sigma$. Let $\sigma'$ be the new schedule obtained from $\sigma$ by exchanging the positions of $J^{(B)}_{y'}$ and $J^{(A)}_x$. From Lemma 3.4(i), we have $C^{(B)}_{y'}(\sigma') = C^{(A)}_x(\sigma) = \sum_{k=1}^{y} p^{(B)}_{k'} + xp^{(A)} \leq \sum_{k=1}^{y} p^{(B)}_{k'} + (y-1)p^{(A)} \leq d^{(B)}_{y'}$. This implies that $J^{(B)}_{y'}$ is also on time in $\sigma'$, and so $\sum U^{(B)}_j(\sigma') = \sum U^{(B)}_j(\sigma) = n$. But $\sum C^{(A)}_j(\sigma') < \sum C^{(A)}_j(\sigma)$. This contradicts the definition of $\sigma$. Property (iii) follows.

To prove property (iv), we note from property (iii) that

$$J^{(B)}_{(j-1)'} \prec_\sigma J^{(A)}_j \prec_\sigma J^{(B)}_{(j+1)'} \text{ and } J^{(A)}_{j-1} \prec_\sigma J^{(B)}_{j'} \prec_\sigma J^{(A)}_{j+1}. \tag{17}$$

By contradiction, suppose that there is some index $j \in \{1, 2, \ldots, n\}$ such that the triple $\{J^{(B)}_{2j-1}, J^{(A)}_j, J^{(B)}_{2j}\}$ violates property (iv).

Let $S^{(j)}(\sigma) = \min\{S^{(B)}_{2j-1}, S^{(A)}_j, S^{(B)}_{2j}\}$ be the earliest starting time for the three jobs $J^{(B)}_{2j-1}, J^{(A)}_j, J^{(B)}_{2j}$ in $\sigma$. From (17), $J^{(B)}_{k'}$ and $J^{(A)}_k$, $1 \leq k \leq j-1$, are just all the jobs completed by time $S^{(j)}(\sigma)$, implying that $S^{(j)}(\sigma) = \sum_{k=1}^{j-1} p^{(B)}_{k'} + (j-1)p^{(A)}$. Thus, we have

$$\sum_{k=1}^{j-1} p^{(B)}_{2k-1} + (j-1)p^{(A)} \leq S^{(j)}(\sigma) \leq \sum_{k=1}^{j-1} p^{(B)}_{2k} + (j-1)p^{(A)}. \tag{18}$$

Suppose that $j' = 2j-1$ and $J^{(A)}_j$ is scheduled before $J^{(B)}_{2j-1}$. Then we have

$$\begin{aligned} C^{(B)}_{2j-1}(\sigma) &= S^{(j)}(\sigma) + p^{(A)} + p^{(B)}_{2j-1} \\ &\geq \sum_{k=1}^{j-1} p^{(B)}_{2k-1} + (j-1)p^{(A)} + p^{(A)} \\ &\quad + M_j + a_{2j-1} \qquad \text{(From (18))} \\ &= \sum_{k=1}^{j-1} (M_k + a_{2k-1}) + p^{(A)} + (j-1)p^{(A)} \\ &\quad + M_j + a_{2j-1} \\ &\geq \sum_{k=1}^{j-1} (M_k + a_{2k}) + 1 + (j-1)p^{(A)} \\ &\quad + M_j + a_{2j-1} \\ &= d^{(B)}_{2j-1} + 1 \\ &> d^{(B)}_{2j-1}. \end{aligned}$$

This contradicts our assumption that $J^{(B)}_{2j-1} = J^{(B)}_{j'}$ is on time in $\sigma$.

From (17) and the choice of $j$, the above discussion implies that $j' = 2j$ and $J^{(A)}_j$ is scheduled after $J^{(B)}_{2j}$. Let $\sigma'$ be the new schedule obtained from $\sigma$ by exchanging the positions of $J^{(A)}_j$ and $J^{(B)}_{2j}$. From property (iii), the completion times for other jobs except $J^{(A)}_j$ and $J^{(B)}_{2j}$ in $\sigma'$ are the same as they are in $\sigma$. From (18), we have $C^{(B)}_{2j}(\sigma') = S^{(j)}(\sigma) + p^{(A)} + p^{(B)}_{2j} \leq \sum_{k=1}^{j} p^{(B)}_{2k} + jp^{(A)} = d^{(B)}_{2j}$. Then $J^{(B)}_{2j}$ is still on time in $\sigma'$, and so $\sum U^{(B)}_j(\sigma') = \sum U^{(B)}_j(\sigma)$. But then $\sum C^{(A)}_j(\sigma') < \sum C^{(A)}_j(\sigma)$. This contradicts the choice of $\sigma$. Property (iv) follows. □

Given a nice sequence $(1', 2', \ldots, n')$, let $(1'', 2'', \ldots, n'')$ be the sequence such that $\{j', j''\} = \{2j-1, 2j\}$ for $j = 1, 2, \ldots, n$. We then define a schedule $\sigma$ of $\mathcal{J}$ corresponding to the nice sequence $(1', 2', \ldots, n')$ in the following way:

$$\begin{aligned} &\{J^{(A)}_1, J^{(B)}_{1'}\} \prec_\sigma \{J^{(A)}_2, J^{(B)}_{2'}\} \prec_\sigma \cdots \prec_\sigma \{J^{(A)}_n, J^{(B)}_{n'}\} \\ &\quad \prec_\sigma J^{(B)}_R \prec_\sigma \{J^{(B)}_{j''} : j = 1, 2, \ldots, n\}, \end{aligned}$$

where, for $j = 1, 2, \ldots, n$, we have

$$\begin{cases} J^{(A)}_j \prec_\sigma J^{(B)}_{j'}, \text{ if } j' = 2j, \\ J^{(B)}_{j'} \prec_\sigma J^{(A)}_j, \text{ if } j' = 2j-1. \end{cases}$$

For $j = 1, 2, \ldots, n$, we have

$$C^{(A)}_j(\sigma) = \begin{cases} jp^{(A)} + \sum_{k=1}^{j} p^{(B)}_{k'}, \text{ if } j' = 2j-1, \\ jp^{(A)} + \sum_{k=1}^{j-1} p^{(B)}_{k'}, \text{ if } j' = 2j. \end{cases}$$

Hence, we have

$$\begin{aligned} &\sum_{j=1}^{n} C^{(A)}_j(\sigma) \\ &\quad = \sum_{j=1}^{n} jp^{(A)} + \sum_{j:j'=2j-1} \sum_{k=1}^{j} p^{(B)}_{k'} \\ &\qquad + \sum_{j:j'=2j} \sum_{k=1}^{j-1} p^{(B)}_{k'} \end{aligned}$$

$$\begin{aligned}
&= \frac{n(n+1)}{2} p^{(A)} + \sum_{j:j'=2j-1} (n+1-j) p^{(B)}_{2j-1} \\
&\quad + \sum_{j:j'=2j} (n-j) p^{(B)}_{2j} \\
&= \frac{n(n+1)}{2} p^{(A)} + \sum_{j=1}^{n} (n-j) p^{(B)}_{2j} \\
&\quad + \sum_{j:j'=2j-1} p^{(B)}_{2j-1} \\
&\quad - \sum_{j:j'=2j-1} (n-j) \left( p^{(B)}_{2j} - p^{(B)}_{2j-1} \right) \\
&= \frac{n(n+1)}{2} p^{(A)} + \sum_{j=1}^{n} (n-j) p^{(B)}_{2j} \\
&\quad + \sum_{j:j'=2j-1} (M_j + a_{2j-1}) \\
&\quad - \sum_{j:j'=2j-1} (n-j)(a_{2j} - a_{2j-1}) \\
&= \frac{n(n+1)}{2} p^{(A)} + \sum_{j=1}^{n} (n-j) p^{(B)}_{2j} \\
&\quad + \sum_{j:j'=2j-1} 3H(a_{2j} - a_{2j-1}).
\end{aligned}$$

From the definition of $Y$ in (4), we have

$$\sum_{j=1}^{n} C^{(A)}_j(\sigma) = Y - 3H\delta + \sum_{j:j'=2j-1} 3H(a_{2j} - a_{2j-1}). \tag{19}$$

Now let us return to the NP-hardness proof. We need to show in the following that the partition instance $\mathcal{I} = (a_1, a_2, \ldots, a_{2n}, H)$ has a solution if and only if the job instance $\mathcal{J}$ has an efficient schedule. Note that $M_j = 3H(a_{2j} - a_{2j-1}) + (n-j)(a_{2j} - a_{2j-1}) - a_{2j-1}$. Then we have

$$M_j + a_{2j-1} - (n-j)(a_{2j} - a_{2j-1}) = 3H(a_{2j} - a_{2j-1}) \quad \text{for } j = 1, 2, \ldots, n. \tag{20}$$

Suppose first that the partition instance $\mathcal{I}$ has a solution. Then there is a partition $(I_1, I_2)$ of the index set $\{1, 2, \ldots, 2n\}$ such that $|I_1 \cap \{2j-1, 2j\}| = |I_2 \cap \{2j-1, 2j\}| = 1$ for $j = 1, 2, \ldots, n$, and $\sum_{j \in I_1} a_j = \sum_{j \in I_2} a_j = H$. For our purpose, we write $I_1 = \{1', 2', \ldots, n'\}$ and $I_2 = \{1'', 2'', \ldots, n''\}$ such that $\{j', j''\} = \{2j-1, 2j\}$ for $j = 1, 2, \ldots, n$. Then $\sum_{k=1}^{n} a_{k'} = \sum_{k=1}^{n} a_{k''} = H$. Recall that $\delta = \frac{1}{2} \sum_{j=1}^{n} (a_{2j} - a_{2j-1})$. Then $2\delta = \sum_{j:j'=2j-1} (a_{2j} - a_{2j-1}) + \sum_{j:j'=2j} (a_{2j} - a_{2j-1})$ and $\sum_{j:j'=2j-1} (a_{2j} - a_{2j-1}) - \sum_{j:j'=2j} (a_{2j} - a_{2j-1}) = \sum_{j=1}^{n} a_{j''} - \sum_{j=1}^{n} a_{j'} = H - H = 0$. Therefore, we have

$$\delta = \sum_{j:j'=2j-1} (a_{2j} - a_{2j-1}). \tag{21}$$

Let $\sigma$ be the schedule of $\mathcal{J}$ corresponding to the nice sequence $(1', 2', \ldots, n')$.

From (19) and (21), we have $\sum C^{(A)}_j(\sigma) = Y$.

For each $j \in \{1, 2, \ldots, n\}$, from Lemma 3.4(i), we have $C^{(B)}_{j'}(\sigma) = \sum_{k=1}^{j} p^{(B)}_{k'} + \lfloor j'/2 \rfloor p^{(A)} \le d^{(B)}_{j'}$. Hence, the $n$ $B$-jobs $J^{(B)}_{j'}$, $1 \le j \le n$, are on time in $\sigma$. For the restricted $B$-job $J^{(B)}_R$, from Lemma 3.4(ii) and from the fact that $\sum_{k=1}^{n} a_{k'} = H$, we have $C^{(B)}_R(\sigma) = \sum_{k=1}^{n} p^{(B)}_{k'} + np^{(A)} + p^{(B)}_R \le d^{(B)}_R$. Thus, $J^{(B)}_R$ is also on time in $\sigma$. Since $J^{(B)}_R$ has a sufficiently large processing time, it is easy to verify that all the $B$-jobs in $\{J^{(B)}_{j''} : j = 1, 2, \ldots, n\}$ are tardy in $\sigma$. It follows that $\sum U^{(B)}_j(\sigma) = n$. Thus, $\sigma$ is an efficient schedule for instance $\mathcal{J}$.

Conversely, suppose that $\sigma$ is an efficient schedule for instance $\mathcal{J}$. Then, $\sum C^{(A)}_j(\sigma) \le Y$ and $\sum U^{(B)}_j(\sigma) = n$. For our purpose, we may assume that $\sigma$ is chosen such that $\sum C^{(A)}_j(\sigma)$ is as small as possible. Recall that $\mathcal{O}(\sigma)$ is the set of on-time $B$-jobs in $\sigma$. According to Lemma 3.6, $J^{(B)}_R$ is on time in $\sigma$. Then $|\mathcal{O}(\sigma) \cap \{J^{(B)}_1, J^{(B)}_2, \ldots, J^{(B)}_{2n}\}| = n$. Assume that $\mathcal{O}(\sigma) \cap \{J^{(B)}_1, J^{(B)}_2, \ldots, J^{(B)}_{2n}\} = \{J^{(B)}_{j'} : 1 \le j \le n\}$ such that $1' < 2' < \cdots < n'$. From Lemma 3.6, we have $j' \in \{2j-1, 2j\}$ for $j = 1, 2, \ldots, n$, and moreover, $(1', 2', \ldots, n')$ is a nice sequence and $\sigma$ is the schedule of $\mathcal{J}$ corresponding to $(1', 2', \ldots, n')$.

Set $I_1 = \{j' : 1 \le j \le n\}$ and $I_2 = \{1, 2, \ldots, 2n\} \backslash I_1$. We will show that $(I_1, I_2)$ is a solution for the partition instance $\mathcal{I}$.

Since $J^{(B)}_R$ is on time in $\sigma$, we have $d^{(B)}_R \ge C^{(B)}_R = np^{(A)} + \sum_{j=1}^{n} p^{(B)}_{j'} + p^{(B)}_R$. From Lemma 3.4(ii), we have

$$\sum_{j \in I_1} a_j = \sum_{j=1}^{n} a_{j'} \le H. \tag{22}$$

Since $\sum C^{(A)}_j(\sigma) \le Y$, from (19), we have $Y \ge \sum C^{(A)}_j(\sigma) = Y - 3H\delta + \sum_{j:j'=2j-1} 3H(a_{2j} - a_{2j-1})$, implying that $\sum_{j:j'=2j-1} (a_{2j} - a_{2j-1}) \le \delta$. Note that $\delta = \frac{1}{2} \sum_{j=1}^{n} (a_{2j} - a_{2j-1})$. Then we have $\sum_{j:j'=2j-1} (a_{2j} - a_{2j-1}) \le \sum_{j:j'=2j} (a_{2j} - a_{2j-1})$, or equivalently,

$$\sum_{j \in I_2} a_j \le \sum_{j \in I_1} a_j. \tag{23}$$

Since $\sum_{j\in I_1} a_j + \sum_{j\in I_2} a_j = 2H$, from (22) and (23), we conclude that $\sum_{j\in I_1} a_j = \sum_{j\in I_2} a_j = H$. Consequently, $(I_1, I_2)$ is a solution for the partition instance $\mathcal{I}$. The result follows. □

We use 1||Lex $(\gamma_1, \gamma_2)$ to denote the single machine hierarchical optimization problem, where $\gamma_1$ is the primary criterion and $\gamma_2$ is the secondary criterion. The objective is to minimize the secondary criterion $\gamma_2$ under the constraint that the primary criterion $\gamma_1$ is optimized.

From the above discussion, we have the following corollary.

**Corollary 3.1** *Problem* $1|p_j^{(A)} = p^{(A)}|Lex\ (\sum U_j^{(B)}, \sum C_j^{(A)})$ *is binary NP-hard.*

**Acknowledgements** The authors would like to thank the associate editor and one referee for their constructive comments and helpful suggestions. This research was supported by NSFC (11671368) and NSFC (11771406).

## Appendix: Proof of Observation 2.2

***Proof*** The proof of Lemma 1 in Leung et al. (2010b) is based on the assumption that there exists a solution for the given instance of the even–odd partition, that is, there is a partition $(I_1, I_2)$ of the index set $\{1, 2, \ldots, 2n\}$ with $|I_1 \cap \{2i-1, 2i\}| = 1$ and $|I_2 \cap \{2i-1, 2i\}| = 1$ for each $i = 1, 2, \ldots, n$ such that $\sum_{j\in I_1} a_j = \sum_{j\in I_2} a_j = H$. From Observation 2.1, the condition $\sum_{j\in I_1} a_j = \sum_{j\in I_2} a_j = H$ is equivalent to condition (C1). The aim of their Lemma 1 is to show that there is a schedule such that $\sum C_j^a \le TC$ and $\sum U_j^b \le n$. This is achieved by constructing a schedule $\pi$.

To guarantee that their constructed schedule $\pi$ satisfies $\sum U_j^b(\pi) \le n$, the $R$-job must be on time in the schedule. Then we have

$$\begin{aligned}
&\sum_{i:2i-1\in I_1} p_{2i-1} + \sum_{i:2i\in I_1} p_{2i} + \sum_{i=1}^{n} x_i + L \\
&\quad = \sum_{i:2i-1\in I_1} a_{2i-1} + \sum_{i:2i\in I_1} (a_{2i} + (l_i - 1)\sigma_i) \\
&\qquad + \sum_{i=1}^{n} x_i + L \\
&\quad = \sum_{i=1}^{n} x_i + H + L + \sum_{i:2i\in I_1} (l_i - 1)\sigma_i \\
&\quad = \sum_{i=1}^{n} x_i + H + L + \sum_{i:2i\in I_1} l_i\sigma_i \\
&\qquad - \sum_{i:2i\in I_1} \sigma_i + \sum_{i:2i-1\in I_1} l_i\sigma_i - \sum_{i:2i-1\in I_1} l_i\sigma_i \\
&\quad = \sum_{i=1}^{n} x_i + H + L + \sum_{i=1}^{n} l_i\sigma_i \\
&\qquad - \sum_{i:2i-1\in I_1} l_i\sigma_i - \sum_{i:2i\in I_1} \sigma_i \\
&\quad = \sum_{i=1}^{n} x_i + H + L + \sum_{i=1}^{n} l_i\sigma_i \\
&\qquad - \sum_{i:2i-1\in I_1} l_i\sigma_i - \frac{1}{2}\sum_{i=1}^{n} \sigma_i \\
&\quad \le d_R = \sum_{i=1}^{n} x_i + \left[ H + \frac{1}{2}\sum_{i=1}^{n} (l_i - 1)\sigma_i \right] + L.
\end{aligned}$$

Thus, we have $\sum_{i:2i-1\in I_1} l_i\sigma_i \ge \frac{1}{2}\sum_{i=1}^{n} l_i\sigma_i$.

On the other hand, to guarantee that the total completion time of agent $A$ is no more than $TC$, we should have

$$\begin{aligned}
&\sum_{i:2i-1\in I_1} (n - i + 1)p_{2i-1} + \sum_{i:2i\in I_1} (n - i)p_{2i} \\
&\qquad + \sum_{i=1}^{n} (n - i + 1)x_i \\
&\quad = \sum_{i:2i-1\in I_1} (n - i + 1)a_{2i-1} + \sum_{i:2i\in I_1} (n - i)(a_{2i} \\
&\qquad + (l_i - 1)\sigma_i) + \sum_{i=1}^{n} (n - i + 1)x_i \\
&\qquad + \sum_{i:2i-1\in I_1} (n - i)(a_{2i} + (l_i - 1)\sigma_i) \\
&\qquad - \sum_{i:2i-1\in I_1} (n - i)(a_{2i} + (l_i - 1)\sigma_i) \\
&\quad = \sum_{i=1}^{n} (n - i)(a_{2i} + (l_i - 1)\sigma_i) \\
&\qquad + \sum_{i=1}^{n} (n - i + 1)x_i + \sum_{i:2i-1\in I_1} a_{2i-1} \\
&\qquad - \sum_{i:2i-1\in I_1} (n - i)(a_{2i} - a_{2i-1} + (l_i - 1)\sigma_i) \\
&\quad = \sum_{i=1}^{n} (n - i)(a_{2i} + (l_i - 1)\sigma_i) + \sum_{i=1}^{n} (n - i + 1)x_i \\
&\qquad + \sum_{i:2i-1\in I_1} (a_{2i-1} - (n - i)l_i\sigma_i) \\
&\quad = \sum_{i=1}^{n} (n - i)(a_{2i} + (l_i - 1)\sigma_i) \\
&\qquad + \sum_{i=1}^{n} (n - i + 1)x_i + \sum_{i:2i-1\in I_1} l_i\sigma_i
\end{aligned}$$

$$\le TC = \sum_{i=1}^{n}(n-i)(a_{2i}+(l_i-1)\sigma_i)$$
$$+\sum_{i=1}^{n}(n-i+1)x_i + \frac{1}{2}\sum_{i=1}^{n} l_i\sigma_i .$$

Thus, we have $\sum_{i:2i-1\in I_1} l_i\sigma_i \le \frac{1}{2}\sum_{i=1}^{n} l_i\sigma_i$.

From the above discussion, we can see that the correctness of their Lemma 1 needs the correctness of condition (C2), i.e., $\sum_{i:2i-1\in I_1} l_i\sigma_i = \sum_{i:2i-1\in I_2} l_i\sigma_i = \frac{1}{2}\sum_{i=1}^{n} l_i\sigma_i$. Consequently, the correctness of their Lemma 1 requires that

Condition (C1) implies condition (C2). $(A)$

The proof of Lemma 10 in Leung et al. (2010b) is based on the assumption that there exists a solution for problem $1||\sum C_j^a \le TC : \sum U_j^b \le n$ on the constructed scheduling instance. The aim is to show that there exists a solution for the instance of the even–odd partition.

Their proof also uses the results in their Lemmas 2–9. We do not need to state all of the steps, but the $R$-job must be on time, and the total completion time for agent $A$ is no more than $TC$. Then, from Lemmas 2–9, we have

$$\sum_{i=1}^{n} p_{2i} + \sum_{i=1}^{n} x_i + L - \sum_{i:P_{2i-1}\text{ is early}} l_i\sigma_i$$
$$= \sum_{i=1}^{n}(a_{2i}+(l_i-1)\sigma_i) + \sum_{i=1}^{n} x_i + L$$
$$- \sum_{i:P_{2i-1}\text{ is early}} l_i\sigma_i$$
$$= H + \frac{1}{2}\sum_{i=1}^{n}\sigma_i + \sum_{i=1}^{n}(l_i-1)\sigma_i + \sum_{i=1}^{n} x_i + L$$
$$- \sum_{i:P_{2i-1}\text{ is early}} l_i\sigma_i$$
$$= \sum_{i=1}^{n} x_i + H + L + \sum_{i=1}^{n} l_i\sigma_i - \frac{1}{2}\sum_{i=1}^{n}\sigma_i$$
$$- \sum_{i:P_{2i-1}\text{ is early}} l_i\sigma_i$$
$$\le d_R = \sum_{i=1}^{n} x_i + \left[H + \frac{1}{2}\sum_{i=1}^{n}(l_i-1)\sigma_i\right] + L,$$

and

$$\sum_{i=1}^{n}(n-i)p_{2i} + \sum_{i=1}^{n}(n-i+1)x_i$$
$$+ \sum_{i:P_{2i-1}\text{ is early}} l_i\sigma_i$$
$$= \sum_{i=1}^{n}(n-i)(a_{2i}+(l_i-1)\sigma_i)$$
$$+ \sum_{i=1}^{n}(n-i+1)x_i + \sum_{i:P_{2i-1}\text{ is early}} l_i\sigma_i$$
$$\le TC = \sum_{i=1}^{n}(n-i)(a_{2i}+(l_i-1)\sigma_i)$$
$$+ \sum_{i=1}^{n}(n-i+1)x_i + \frac{1}{2}\sum_{i=1}^{n} l_i\sigma_i .$$

Hence, we have that

$$\sum_{i:P_{2i-1}\text{ is early}} l_i\sigma_i \ge \frac{1}{2}\sum_{i=1}^{n} l_i\sigma_i \text{ and } \sum_{i:P_{2i-1}\text{ is early}} l_i\sigma_i \le \frac{1}{2}\sum_{i=1}^{n} l_i\sigma_i .$$

That is,

$$\sum_{i:P_{2i-1}\text{ is early}} l_i\sigma_i = \sum_{i:P_{2i-1}\text{ is tardy}} l_i\sigma_i = \frac{1}{2}\sum_{i=1}^{n} l_i\sigma_i . \quad (C2')$$

Note that condition (C2′) is potentially implied in the proof of their Lemma 10. After this, the authors of Leung et al. (2010b) directly conclude that the instance of the even–odd partition has a solution. In the normal deduction, we should add the following procedure.

Let $I_1 = \{2i-1 : P_{2i-1} \text{ is early}\} \cup \{2i : P_{2i} \text{ is early}\}$ and $I_2 = \{1, 2, \ldots, 2n\} \setminus I_1$. The results in their Lemmas 2-9 imply that $|I_1 \cap \{2i-1, 2i\}| = 1$ and $|I_2 \cap \{2i-1, 2i\}| = 1$ for each $i = 1, 2, \ldots, n$. Now condition (C2′) can be rewritten as condition (C2), i.e., $\sum_{i:2i-1\in I_1} l_i\sigma_i = \sum_{i:2i-1\in I_2} l_i\sigma_i = \frac{1}{2}\sum_{i=1}^{n} l_i\sigma_i$. Thus, from Observation 2.1, the correctness of their Lemma 10 requires that

Condition (C2) implies condition (C1). $(A)$

From equations (A) and (B), we conclude that if their Lemmas 1 and 10 are correct, then for every partition $(I_1, I_2)$ of the index set $\{1, 2, \ldots, 2n\}$ with $|I_1 \cap \{2i-1, 2i\}| = 1$ and $|I_2 \cap \{2i-1, 2i\}| = 1$ for each $i = 1, 2, \ldots, n$, the two conditions (C1) and (C2) are equivalent. □

## References

Agnetis, A., Mirchandani, P. B., Pacciarelli, D., & Pacifici, A. (2004). Scheduling problems with two competing agents. *Operations Research*, *52*, 229–242.

Baker, K. R., & Smith, J. C. (2003). A multiple-criterion model for machine scheduling. *Journal of Scheduling*, *6*, 7–16.

Clifford, J. J., & Posner, M. E. (2001). Parallel machine scheduling with high multiplicity. *Mathematical Programming*, *89*, 359–383.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. San Francisco: Freeman.

Graham, R. L., Lawler, E. L., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1979). Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, *5*, 287–326.

Hochbaum, D. S., & Shamir, R. (1991). Strongly polynomial algorithms for the high multiplicity scheduling problem. *Operations Research*, *39*, 648–653.

Leung, J. Y. T., Pinedo, M., & Wan, G. H. (2010a). Competitive two-agent scheduling and its applications. *Operations Research*, *58*, 458–469.

Leung, J. Y. T., Pinedo, M., & Wan, G. H. (2010b). Electronic Companion—"Competitive two-agent scheduling and its applications". *Operations Research*, https://doi.org/10.1287/opre.1090.0744ec.

Moore, J. M. (1968). An n job, one machine sequencing algorithm for minimizing the number of late jobs. *Management Science*, *15*, 102–109.

Ng, C. T., Cheng, T. C. E., & Yuan, J. J. (2006). A note on the complexity of the problem of two-agent scheduling on a single machine. *Journal of Combinatorial Optimization*, *12*, 387–394.