mathematics

Article

Transportation and Batching Scheduling for Minimizing Total Weighted Completion Time

Hongjun Wei *, Jinjiang Yuan and Yuan Gao

School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001, China
* Correspondence: weihongjun@zzu.edu.cn

Received: 20 August 2019; Accepted: 3 September 2019; Published: 5 September 2019

Abstract: We consider the coordination of transportation and batching scheduling with one single vehicle for minimizing total weighted completion time. The computational complexity of the problem with batch capacity of at least 2 was posed as open in the literature. For this problem, we show the unary NP-hardness for every batch capacity at least 3 and present a polynomial-time 3-approximation algorithm when the batch capacity is at least 2.

Keywords: Transportation; batching scheduling; total weighted completion time; unary NP-hard; approximation algorithm

---

1. Introduction

Tang and Gong [1] first raised and studied the problem of transportation and batching scheduling (TBS). This model, which combines transportation and scheduling together, is motivated by a production environment in which a set of semi-finished jobs are transported from a holding area to a manufacturing facility for further processing by the available transporters and is used in many manufacturing systems. This is particularly true in the iron and steel industry.

Formally, we can describe the TBS problem in the following way. We are given a set of jobs, a set of vehicles (transporters), and a single batching machine that can handle batch jobs at the same time. Initially, all jobs and all vehicles are located at a holding area and available from time zero onward. When the production process begins, all jobs have to be transported by the vehicles to the batching machine and further processing is then carried out, where each vehicle can deliver one job at a time. The transportation time of a job is job-dependent, the empty moving times of the vehicles from the batching machine back to the holding area are identical, and the processing times of the batches on the batching machine are identical. The following notations are used in this scheduling model.

- $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$ is a set of n jobs to be processed.
- $M = \{1, 2, \ldots, m\}$ is a set of m vehicles used to transport the jobs.
- τ_j is the transportation time of job J_j, $j = 1, 2, \ldots, n$, from the holding area to the batching machine.
- τ is the empty moving time of each vehicle from the machine back to the holding area. In the sequel, we simply call τ the vehicle return time.
- c is the capacity of the batching machine. We require that every batch consists at most c jobs.
- p is the processing time of each batch, which is independent of the jobs composing the batch.
- C_j is the completion time of jobs J_j in a schedule, $j = 1, 2, \ldots, n$.
- $\alpha(b)$, which is an increasing function in b, is the processing (or batching) cost if a total of b batches are generated in of a schedule. The more batches there are, the more processing costs there will be.
- f is the scheduling cost which depends on the completion times $C_1, C_2, \ldots, C_n$ of the jobs.

The goal of the TBS problem is to find a feasible schedule that minimizes the scheduling cost plus the processing cost. We will denote this problem by $(m, c)|\tau|f + \alpha(b)$, where "$(m, c)$" means that we have m vehicles in the transportation stage and the batching machine has a capacity c for forming a batch and "τ" indicates the vehicle return time.

Production–transportation problems, which have some similarities as the TBS problems, have also been extensively studied in the literature. Hall and Potts [2] introduced and studied various single-machine and parallel-machine scheduling problems in which the various cost functions being considered are based on the delivery times and delivery cost. Chen [3] surveyed the existing models of integrated production and outbound distribution scheduling (IPODS) and presented a unified model representation method. The author also classified the existing models into several different classes and provided an overview for the optimality properties, computational tractability, and solution algorithms. As mentioned by Tang and Gong [1], the TBS problem differs from the IPODS problem. In fact, in the TBS setting, apart from the schedule of the semi-finished jobs in the transportation stage, we also consider the schedule of these jobs on the batching machine in the production stage.

Recall that a combinatorial optimization problem is binary (unary) NP-hard if it is NP-hard in the binary (unary) encoding.

Tang and Gong [1] studied the TBS problem which aims to minimize the sum of the total completion time of the jobs and the processing cost of the batching machine. For this problem, the authors proved the binary NP-hardness and further established a pseudo-polynomial-time algorithm and an FPTAS for any fixed m.

For the more "classic" scheduling objectives that exclude the processing cost, Zhu et al. [4] showed that the complexity result in Tang and Gong [1] is still valid, that is, the TBS problem with fixed m for minimizing the total completion time of the jobs is binary NP-hard. When m is arbitrary, Zhu et al. [4] showed that the TBS problem for minimizing the total completion time of the jobs is unary NP-hard. Moreover, they proved that the TBS problem for minimizing the sum of the total weighted completion time of the jobs and the processing cost of the batching machine is unary NP-hard even if $m = 1$ and $c = 3$. The computational complexity of the TBS problem with $m = 1$ for just minimizing the total weighted completion time of the jobs was posed as an open problem in Zhu et al. [4]. It should be noticed that, in the case where $\tau = 0$, i.e., the vehicle return time is given by 0, the model of transportation times in the TBS problems can be considered as a special case of setup times studied in Allahverdi [5] and Ciavotta et al., [6]

In this paper, we consider the TBS problem $(1, c)|\tau| \sum w_j C_j$, in which we have one single vehicle in the transportation stage, the scheduling criterion is to minimize the total weighted completion time of the jobs, and the processing cost is given by 0.

Note that when $c = 1$ and $\tau = 0$, problem $(1, c)|\tau| \sum w_j C_j$ degenerates to the classical two-machine flow-shop scheduling problem $F2|p_{2j} = p| \sum w_j C_j$. Recently, Wei and Yuan [7] showed that problem $F2|p_{2j} = p| \sum w_j C_j$ is unary NP-hard and admits a 2-approximation algorithm. More research of problem $F2|| \sum w_j C_j$ can be found in Choi et al. [8] and Hoogeveen and Kawaguchi [9].

The unary NP-hardness of problem $F2|p_{2j} = p| \sum w_j C_j$, established in the work by the authors of [7], implies that problem $(1, 1)|\tau = 0| \sum w_j C_j$ is unary NP-hard. However, in general, the computational complexity of problem $(1, c)|\tau| \sum w_j C_j$ for $c \geq 2$ is unaddressed.

In this paper, first, we show that for every $c \geq 3$ (including the possibility $c = n$), problem $(1, c)|\tau = 0| \sum w_j C_j$ is unary NP-hard. Then, for the general problem $(1, c)|\tau| \sum w_j C_j$ with the batch capacity $c \geq 2$, we present a polynomial-time approximation algorithm, which has a worst-case performance ratio of less than 3. The complexity of problem $(1, 2)|\tau| \sum w_j C_j$ is still open.

2. Unary NP-Hardness Proof

To show the unary NP-hardness of problem $(1, c)|\tau = 0| \sum w_j C_j$ with $c \geq 3$, we will use the following decision problem "3-Partition" as the source problem. As outlined by Garey and Johnson [10], a 3-partition is unary NP-complete.

3-Partition: In an instance of the problem, we are given a set $\{a_1, a_2, \ldots, a_{3t}, B\}$ of $3t + 1$ positive integers satisfying $\frac{1}{4}B < a_j < \frac{1}{2}B$ for $j = 1, 2, \ldots, 3t$ and $\sum_{j=1}^{3t} a_j = tB$. The decision asks, is there a partition of the index set $I = \{1, 2, \ldots, 3t\}$ into t parts $I_1, I_2, \ldots, I_t$ such that $|I_i| = 3$ and $\sum_{j \in I_i} a_j = B$ for all $i = 1, 2, \ldots, t$?

The following useful lemma states a basic algebraic result.

Lemma 1. *Let $x_1, x_2, \ldots, x_k$ be k positive numbers. Then $\sum_{i=1}^{k} x_i^2 \geq k \cdot \left(\frac{x_1 + x_2 + \cdots + x_k}{k}\right)^2$, and moreover the equality holds if and only if $x_1 = x_2 = \cdots = x_k = \frac{x_1 + x_2 + \cdots + x_k}{k}$.*

Theorem 1. *For every $c \geq 3$, problem $(1, c)|\tau = 0|\sum w_j C_j$ is unary NP-hard.*

Proof. For a given instance $(a_1, a_2, \ldots, a_{3t}; B)$ of 3-Partition, we first define

$$\Delta = t^2 B^2 + 1 = O(t^2 B^2) \tag{1}$$

and

$$M = t(t + 3)(3\Delta + B)^2 = O(t^6 B^4). \tag{2}$$

Then we construct a scheduling instance of problem $(1, c)|\tau = 0|\sum w_j C_j$ as follows.

- There are $n = 3t + 1$ jobs $J_0, J_1, \ldots, J_{3t}$ of two types:
  - (i) J_0, called the 0-job, has a transportation time $\tau_0 = 0$ and a weights $w_0 = M$, and
  - (ii) $J_1, J_2, \ldots, J_{3t}$, called partition jobs, have transportation times $\tau_j = \Delta + a_j$ and weights $w_j = \Delta + a_j$ for $j = 1, 2, \ldots, 3t$.
- The number of vehicles is given by $m = 1$.
- The vehicle return time is given by $\tau = 0$.
- The batch machine capacity $c \geq 3$ is arbitrary, where $c = n$ is allowed.
- The batch processing time is given by $p = 3\Delta + B = O(t^2 B^2)$.
- The threshold value is given by

$$Q = M(3\Delta + B) + \frac{1}{2}t(t + 3)(3\Delta + B)^2 = Mp + \frac{1}{2}t(t + 3)p^2. \tag{3}$$

The above scheduling instance has $6t + 6$ parameters: τ_j, w_j ($j = 0, 1, \ldots, 3t$), τ, p, c, and Q, with Q being the largest one. Since $M = O(t^6 B^4)$ and $p = O(t^2 B^2)$, from Equation (3), we have $Q = Mp + \frac{1}{2}t(t + 3)p^2 = O(t^8 B^6)$. This implies that the size of the above scheduling instance under the unary encoding is upper bounded by $O(t^9 B^6)$. Note that the size of the 3-partition instance under the unary encoding is given by $O(tB)$. Then, our scheduling instance can be constructed from the 3-partition instance in a polynomial time under the unary encoding. From the general principle of NP-hardness proof, we need to show in the following that the 3-Partition instance has a solution if and only if the scheduling instance has a feasible schedule π such that $\sum w_j C_j(\pi) \leq Q$.

Let us first suppose that the 3-Partition instance has a solution, which means that there is a partition of the index set $I = \{1, 2, \ldots, 3t\}$ into t parts $I_1, I_2, \ldots, I_t$ such that $|I_i| = 3$ and $\sum_{j \in I_i} a_j = B$ for all $i = 1, 2, \ldots, t$. Let $\mathcal{J}_0 = \{J_0\}$ and $\mathcal{J}_i = \{J_j : j \in I_i\}$ for all $i = 1, 2, \ldots, t$. We define a schedule π for the scheduling instance in the following way.

- The vehicle consecutively transports the $3t + 1$ jobs in the order

$$\mathcal{J}_0 \prec \mathcal{J}_1 \prec \cdots \prec \mathcal{J}_t \tag{4}$$

  one by one, where the transportation order of the three jobs in each $\mathcal{J}_i$, $i = 1, 2, \ldots, t$, does not matter.
- The batching machine takes each $\mathcal{J}_i$, $i = 0, 1, 2, \ldots, t$, as a single batch and processes the $t + 1$ batches in the order described in Equation (4) as they are transported. Then we have totally $t + 1$ processing batches.

Note that the transportation time of the 0-job in $\mathcal{J}_0$ is 0, and the total transportation time of the three partition-jobs in $\mathcal{J}_i$, $i \in \{1, 2, \ldots, t\}$, is given by $\sum_{J_j \in \mathcal{J}_i} \tau_j = \sum_{j \in I_i} (\Delta + a_i) = 3\Delta + B$. From $p = 3\Delta + B$, the completion times of the $t + 1$ batches $\mathcal{J}_0 \prec \mathcal{J}_1 \prec \cdots \prec \mathcal{J}_t$ are given by $p, 2p, \ldots, (t + 1)p$, respectively. Moreover, the weight of the 0-job in $\mathcal{J}_0$ is M and the total weight of the three partition-jobs in $\mathcal{J}_i$, $i \in \{1, 2, \ldots, t\}$, is given by $\sum_{J_j \in \mathcal{J}_i} w_j = \sum_{j \in I_i} (\Delta + a_i) = 3\Delta + B = p$. Then we have

$$\sum w_j C_j(\pi) = M \cdot p + p^2 \cdot (2 + 3 + \cdots + (t + 1)) = Mp + \frac{1}{2} t(t + 3)p^2.$$

From Equation (3), we have $Q = Mp + \frac{1}{2} t(t + 3)p^2$, which leads to the relation $\sum w_j C_j(\pi) = Q$. Therefore, π is a required schedule. This proves the necessity.

We next prove the sufficiency. To this end, we suppose that π is a feasible schedule of the scheduling instance, such that $\sum w_j C_j(\pi) \leq Q$. Recall that $Q = Mp + \frac{1}{2} t(t + 3)p^2$. Let $\mathcal{B}_0, \mathcal{B}_1, \ldots, \mathcal{B}_K$ be the batch sequence processed by the batching machine in π in this order.

If the 0-job J_0 completes after time p, we have $\sum w_j C_j(\pi) > w_0 C_0(\pi) \geq M(p + 1) = Mp + M$. From the definition of M in Equation (2), we have $M = t(t + 3)(3\Delta + B)^2 = t(t + 3)p^2$. Thus, $\sum w_j C_j(\pi) > Mp + t(t + 3)p^2 > Q$, contradicting the choice of π. Consequently, we have

$$C_0(\pi) = p,\ \mathcal{B}_0 = \{J_0\},\ \text{and}\ w_0 C_0(\pi) = Mp. \tag{5}$$

From Equation (5) and from the fact that $\sum w_j C_j(\pi) \leq Q = Mp + \frac{1}{2} t(t + 3)p^2$, we have

$$\sum_{j=1}^{3t} w_j C_j(\pi) \leq \frac{1}{2} t(t + 3)p^2. \tag{6}$$

From the above discussion, we know that the $3t$ partition-jobs $J_1, J_2, \ldots, J_{3t}$ are distributed into the K batches $\mathcal{B}_i$, $i = 1, 2, \ldots, K$. Then we define

$$I_i = \{j : J_j \in \mathcal{B}_i\}\ \text{and}\ A_i = \textstyle\sum_{j \in I_i} a_j,\ i = 1, 2, \ldots, K. \tag{7}$$

For each $i \in \{1, 2, \ldots, K\}$, we define $w^{(i)} = \sum_{j \in I_i} w_j$ and $\tau^{(i)} = \sum_{j \in I_i} \tau_j$. Since $w_j = \tau_j = \Delta + a_j$ for $j \in \{1, 2, \ldots, 3t\}$, we have

$$w^{(i)} = \tau^{(i)} = |I_i|\Delta + A_i,\ i = 1, 2, \ldots, K. \tag{8}$$

Since each batch $\mathcal{B}_i$ cannot be processed before all the jobs in $\mathcal{B}_1 \cup \mathcal{B}_2 \cup \cdots \cup \mathcal{B}_i$ are transported, we have

$$C_{\mathcal{B}_i}(\pi) \geq \tau^{(1)} + \tau^{(2)} + \cdots + \tau^{(i)} + p,\ i = 1, 2, \ldots, K. \tag{9}$$

Since the batches are processed in the order $\mathcal{B}_0 \prec \mathcal{B}_1 \prec \cdots \prec \mathcal{B}_K$ on the batching machine in π, we further have

$$C_{\mathcal{B}_i}(\pi) \geq (i + 1)p,\ i = 1, 2, \ldots, K. \tag{10}$$

Note that $\sum_{i=1}^{K} \tau^{(i)} = \sum_{j=1}^{3t} \tau_j = \sum_{j=1}^{3t} (\Delta + a_j) = 3t\Delta + tB = tp$. We show in the following that $K = t$.

If $K \geq t+1$, let $\tau^* = \tau^{(t+1)} + \tau^{(t+2)} + \cdots + \tau^{(K)}$. Then $\Delta < \tau^* < tp - t\Delta$. From Equations (9) and (10), we have

$$\begin{aligned}
\sum_{j=1}^{3t} w_j C_j(\pi) &= \sum_{i=1}^{K} w^{(i)} C_{\mathcal{B}_i}(\pi) \\
&= \sum_{i=1}^{t} \tau^{(i)} C_{\mathcal{B}_i}(\pi) + \sum_{i=t+1}^{K} \tau^{(i)} C_{\mathcal{B}_i}(\pi) \\
&\geq \sum_{i=1}^{t} \tau^{(i)} (\tau^{(1)} + \tau^{(2)} + \cdots + \tau^{(i)} + p) + \tau^*(t+2)p \\
&= \tfrac{1}{2}(\sum_{i=1}^{t} (\tau^{(i)})^2 + (\sum_{i=1}^{t} \tau^{(i)})^2) + \sum_{i=1}^{t} \tau^{(i)} p + \tau^*(t+2)p \\
&= \tfrac{1}{2} \sum_{i=1}^{t} (\tau^{(i)})^2 + \tfrac{1}{2}(tp-\tau^*)^2 + (tp-\tau^*)p + \tau^*(t+2)p \\
&\geq \tfrac{1}{2} \cdot \tfrac{(tp-\tau^*)^2}{t} + \tfrac{1}{2}(tp-\tau^*)^2 + (tp-\tau^*)p + \tau^*(t+2)p \\
&> \tfrac{1}{2} t(t+3)p^2,
\end{aligned}$$

where the second inequality follows from Lemma 1 and the last inequality follows by a simple calculation. This contradicts the relation in Equation (6).

If $K \leq t-1$, from Lemma 1 and Equation (9), we have

$$\begin{aligned}
\sum_{j=1}^{3t} w_j C_j(\pi) &= \sum_{i=1}^{K} w^{(i)} C_{\mathcal{B}_i}(\pi) \\
&\geq \sum_{i=1}^{K} \tau^{(i)} (\tau^{(1)} + \tau^{(2)} + \cdots + \tau^{(i)} + p) \\
&= \tfrac{1}{2}(\sum_{i=1}^{K} (\tau^{(i)})^2 + (\sum_{i=1}^{K} \tau^{(i)})^2) + \sum_{i=1}^{K} \tau^{(i)} p \\
&= \tfrac{1}{2}(\sum_{i=1}^{K} (\tau^{(i)})^2 + t^2p^2) + tp^2 \\
&\geq \tfrac{1}{2}(t^2p^2/K + t^2p^2) + tp^2 \\
&> \tfrac{1}{2}(tp^2 + t^2p^2) + tp^2 \\
&= \tfrac{1}{2} t(t+3)p^2,
\end{aligned}$$

contradicting the relation in Equation (6) again.

The above discussion shows that $K = t$. Thus, from Lemma 1 and Equation (9) again, we have

$$\begin{aligned}
\sum_{j=1}^{3t} w_j C_j(\pi) &= \sum_{i=1}^{t} w^{(i)} C_{\mathcal{B}_i}(\pi) \\
&\geq \sum_{i=1}^{t} \tau^{(i)} (\tau^{(1)} + \tau^{(2)} + \cdots + \tau^{(i)} + p) \\
&= \tfrac{1}{2}(\sum_{i=1}^{t} (\tau^{(i)})^2 + (\sum_{i=1}^{t} \tau^{(i)})^2) + \sum_{i=1}^{t} \tau^{(i)} p \\
&= \tfrac{1}{2}(\sum_{i=1}^{t} (\tau^{(i)})^2 + t^2p^2) + tp^2 \\
&\geq \tfrac{1}{2}(tp^2 + t^2p^2) + tp^2 \\
&= \tfrac{1}{2} t(t+3)p^2 \\
&\geq \sum_{j=1}^{3t} w_j C_j(\pi),
\end{aligned}$$

where the last inequality follows from Equation (6). This means that all the inequalities in the above deduction must hold with equalities. In particular, we have $\sum_{i=1}^{t} (\tau^{(i)})^2 = tp^2$. From Lemma 1 again, it holds that $\tau^{(1)} = \tau^{(2)} = \cdots = \tau^{(t)} = p = 3\Delta + B$. From Equation (8), we conclude that

$$|I_1|\Delta + A_1 = |I_2|\Delta + A_2 = \cdots = |I_t|\Delta + A_t = 3\Delta + B. \tag{11}$$

Since $A_i = \sum_{j \in I_i} a_j \leq tB$ for all $i = 1, 2, \ldots, t$ and the value $\Delta = t^2B^2 + 1$ defined in (1) is sufficiently large, from Equation (11), we can easily deduce that $|I_i| = 3$ and $A_i = B$ for all $i = 1, 2, \ldots, t$. Consequently, the 3-partition instance has a solution. The result follows. □

3. Approximation

We assume in this section that $c \geq 2$. Given a job instance $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$ of problem $(1,c)|\tau|\sum w_j C_j$, we define

$$p_{1,j} = \tau_j + \tau,\ j = 1, 2, \ldots, n. \tag{12}$$

In $O(n \log n)$ time, we can renumber the n jobs in the nondecreasing order of the ratios $(p_{1j} + p/c)/w_j$ such that

$$(p_{1,1} + p/c)/w_1 \leq (p_{1,2} + p/c)/w_2 \leq \cdots \leq (p_{1,n} + p/c)/w_n. \tag{13}$$

In $O(n \log n)$ time, we can also obtain a permutation $(1', 2', \ldots, n')$ of $\{1, 2, \ldots, n\}$ such that $\tau_{1'} \geq \tau_{2'} \geq \cdots \geq \tau_{n'}$. Then, we define

$$\alpha = \min\{\tau_1, \tau_2, \cdots, \tau_n, \lfloor p/c \rfloor\} \tag{14}$$

and

$$\beta = \max\{p,\ \tau_{1'} + \tau_{2'} + \cdots + \tau_{c'} + c\tau\}. \tag{15}$$

For a schedule π of the job instance $\mathcal{J}$, we use $C_{1,j}(\pi)$ to denote the completion time of job J_j on the vehicle. Since the batch containing J_j must start at or after time $C_{1,j}(\pi)$, we have

$$C_j(\pi) \geq C_{1,j}(\pi) + p,\ j = 1, 2, \ldots, n. \tag{16}$$

By the job-exchanging argument, we can show that there must be an optimal schedule π of problem $(1,c)|\tau|\sum w_j C_j$ such that for the two jobs J_i and J_j,

$$C_{1,i}(\pi) < C_{1,j}(\pi) \Rightarrow C_i(\pi) \leq C_j(\pi). \tag{17}$$

Then we only consider schedules with the property in (17) in the sequel.

In the following we present an approximation algorithm for problem $(1,c)|\tau|\sum w_j C_j$ with the worst-case performance ratio at most $3\beta/(\beta + \alpha) \leq 3$, where $c \geq 2$. Our approximation algorithm can be described in the following way.

Algorithm 1. *For problem $(1,c)|\tau|\sum w_j C_j$ on instance $\mathcal{J}$.*

Step 1. Schedule the jobs in the order $J_1, J_2, \ldots, J_n$ on the vehicle without idle time.
Step 2. Form batches and process them on the batching machine by using the following strategy: When the batching machine is idle at time t and some jobs are available for processing at time t, it forms and process a new batch, which contains as many jobs as possible subject to the batch capacity c, by the rule that jobs with small subscriptions have the priority to be processed.

Clearly, Algorithm 1 runs in $O(n)$ time. To analyze the worst-case performance ratio of Algorithm 1, we first establish a lower bound of the optimal cost of problem $(1,c)|\tau|\sum w_j C_j$ on instance $\mathcal{J}$.

Lemma 2. *Let π^* be an optimal schedule of instance $\mathcal{J}$ and, for each $j \in \{1, 2, \ldots, n\}$, let $J_{[j]}$ denote the job that occupies the jth position on the vehicle in π^*. Then*

$$\sum_{j=1}^{n} w_{[j]} C_{[j]}(\pi^*) \geq \frac{1}{3}\Big(\sum_{j=1}^{n} \sum_{i=1}^{j} w_{[j]}(p_{1,[i]} + p/c) + (3\alpha + 2p - \tau) \sum_{j=1}^{n} w_{[j]} \Big). \tag{18}$$

Proof. Since π^* satisfies the property in (17), we may assume that $C_{[1]}(\pi^*) \le C_{[2]}(\pi^*) \le \cdots \le C_{[n]}(\pi^*)$. Note that $c \ge 2$ is the batch capacity and each batch has a processing time p. Moreover, the first batch on the batching machine starts at a time greater than $\tau_{[1]} = p_{1,[1]} - \tau$. For each $j = 1, 2, \ldots, n$, at least $\lceil j/c \rceil$ batches are completed by time $C_{[j]}(\pi^*)$ on the batch machine. Then we have $C_{[j]}(\pi^*) \ge p_{1,[1]} - \tau + \lceil j/c \rceil p$ for $j = 1, 2, \ldots, n$. Consequently, we have

$$\sum_{j=1}^{n} w_{[j]} C_{[j]}(\pi^*) \ge (p_{1,[1]} - \tau) \sum_{j=1}^{n} w_{[j]} + p \sum_{j=1}^{n} \lceil j/c \rceil w_{[j]}. \tag{19}$$

From (16), for each $j = 1, 2, \cdots, n$, we also have $C_{[j]}(\pi^*) \ge C_{1,[j]}(\pi^*) + p$, which implies that $C_{[j]}(\pi^*) \ge \sum_{i=1}^{j} p_{1,[i]} + p - \tau$. Consequently, we have

$$\sum_{j=1}^{n} w_{[j]} C_{[j]}(\pi^*) \ge \sum_{j=1}^{n} \sum_{i=1}^{j} w_{[j]} p_{1,[i]} + (p - \tau) \sum_{j=1}^{n} w_{[j]}. \tag{20}$$

From the above two inequalities, (19) and (20), we obtain

$$\begin{aligned} 3\sum_{j=1}^{n} w_{[j]} C_{[j]}(\pi^*) &\ge \sum_{j=1}^{n} w_{[j]} \big(p_{1,[1]} - \tau + \lceil j/c \rceil p + 2\sum_{i=1}^{j} p_{1,[i]} + 2(p - \tau) \big) \\ &\ge \sum_{j=1}^{n} w_{[j]} \big(\sum_{i=1}^{j} (p_{1,[i]} + p/c) + 3\alpha + 2p - \tau \big) \\ &= \sum_{j=1}^{n} \sum_{i=1}^{j} w_{[j]} (p_{1,[i]} + p/c) + (3\alpha + 2p - \tau) \sum_{j=1}^{n} w_{[j]}. \end{aligned}$$

Then the lemma follows immediately. □

The following lemma is also useful in our discussion.

Lemma 3. *For any c indices $i_1, i_2, \cdots, i_c \in \{1, 2, \cdots, n\}$, we have*

$$|p_{1,i_1} + p_{1,i_2} + \cdots + p_{1,i_c} - p| \le \frac{\beta - \alpha}{\beta + \alpha} (p_{1,i_1} + p_{1,i_2} + \cdots + p_{1,i_c} + p).$$

Proof. Let $x = \min\{p_{1,i_1} + p_{1,i_2} + \cdots + p_{1,i_c}, p\}$ and $y = \max\{p_{1,i_1} + p_{1,i_2} + \cdots + p_{1,i_c}, p\}$. Then $|p_{1,i_1} + p_{1,i_2} + \cdots + p_{1,i_c} - p| = y - x$ and $p_{1,i_1} + p_{1,i_2} + \cdots + p_{1,i_c} + p = x + y$. From the definitions of α and β in Equations (14) and (15), we further have $\alpha \le x \le y \le \beta$. This implies that $y\alpha \le \beta\alpha \le x\beta$. Now

$$\begin{aligned} & |p_{1,i_1} + p_{1,i_2} + \cdots + p_{1,i_c} - p|(\beta + \alpha) \\ = \; & (y - x)(\beta + \alpha) = y\beta + y\alpha - x\beta - x\alpha \\ \le \; & y\beta - y\alpha + x\beta - x\alpha \\ = \; & (x + y)(\beta - \alpha) \\ = \; & (\beta - \alpha)(p_{1,i_1} + p_{1,i_2} + \cdots + p_{1,i_c} + p). \end{aligned}$$

It follows that $|p_{1,i_1} + p_{1,i_2} + \cdots + p_{1,i_c} - p| \le (\beta - \alpha)(p_{1,i_1} + p_{1,i_2} + \cdots + p_{1,i_c} + p)/(\beta + \alpha)$. □

Now we are ready to establish our final result. Recall that $c \ge 2$.

Theorem 2. *Algorithm 1 yields a schedule with cost no more than $3\beta/(\alpha + \beta)$ times the cost of an optimal schedule.*

Proof. Let π be the schedule of instance $\mathcal{J}$ generated by Algorithm 1. Since the jobs in $\mathcal{J}$ are scheduled in the order $J_1 \prec J_2 \prec \cdots \prec J_n$ on the vehicle without idle time, the implementation of Step 2 implies that, for $j = 1, 2, \ldots, c$, we have

$$\begin{aligned}
C_j(\pi) &\leq \textstyle\sum_{i=1}^{j} p_{1,i} + 2p - \tau \\
&= \beta(\textstyle\sum_{i=1}^{j} p_{1,i} + 2p - \tau)/(\beta+\alpha) + \alpha(\textstyle\sum_{i=1}^{j} p_{1,i} + 2p - \tau)/(\beta+\alpha) \\
&\leq \beta(\textstyle\sum_{i=1}^{j} p_{1,i} + 2p - \tau)/(\beta+\alpha) + 3\alpha\beta/(\beta+\alpha) \\
&\leq \beta(\textstyle\sum_{i=1}^{j} (p_{1,i} + p/c) + 3\alpha + 2p - \tau)/(\beta+\alpha).
\end{aligned}$$

For each $j = c+1, c+2, \ldots, n$, we define $k_j = \lceil j/c \rceil - 1$ and $l_j = j - k_j c$. Then, we have

$$\begin{aligned}
C_j(\pi) &\leq p + \max\{C_{1,j}(\pi) + p,\ C_{j-c}(\pi)\} \\
&\leq p + \max\{C_{1,j-c}(\pi) + p + \textstyle\sum_{h=1}^{c} p_{1,j-c+h},\ \max\{C_{1,j-c}(\pi) + p, C_{j-2c}(\pi)\} + p\} \\
&\leq p + \max\{C_{1,j-c}(\pi) + p, C_{j-2c}(\pi)\} + \max\{\textstyle\sum_{h=1}^{c} p_{1,j-c+h}, p\} \\
&\leq p + \max\{C_{1,l_j+c}(\pi) + p, C_{l_j}(\pi)\} + \textstyle\sum_{i=1}^{k_j-1} \max\{\textstyle\sum_{h=1}^{c} p_{1,l_j+ic+h}, p\} \\
&\leq p + \max\{C_{1,l_j+c}(\pi) + p, C_{1,l_j}(\pi) + 2p\} + \textstyle\sum_{i=1}^{k_j-1} \max\{\textstyle\sum_{h=1}^{c} p_{1,l_j+ic+h}, p\} \\
&\leq 2p + C_{1,l_j}(\pi) + \textstyle\sum_{i=0}^{k_j-1} \max\{\textstyle\sum_{h=1}^{c} p_{1,l_j+ic+h}, p\} \\
&= \textstyle\sum_{i=1}^{l_j} p_{1,i} + 2p - \tau + \textstyle\sum_{i=0}^{k_j-1} \max\{\textstyle\sum_{h=1}^{c} p_{1,l_j+ic+h}, p\}.
\end{aligned}$$

By Lemma 3 and using the algebraic equality

$$2 \cdot \max\{x, y\} = x + y + |x - y|, \text{ for every two real numbers } x \text{ and } y,$$

we can obtain that

$$\begin{aligned}
&\max\{\textstyle\sum_{h=1}^{c} p_{1,l_j+ic+h}, p\} \\
=\ &\tfrac{1}{2}\big(\textstyle\sum_{h=1}^{c} p_{1,l_j+ic+h} + p + |\textstyle\sum_{h=1}^{c} p_{1,l_j+ic+h} - p|\big) \\
\leq\ &\tfrac{1}{2}\big(\textstyle\sum_{h=1}^{c} p_{1,l_j+ic+h} + p + (\textstyle\sum_{h=1}^{c} p_{1,l_j+ic+h} + p)(\beta-\alpha)/(\beta+\alpha)\big) \\
=\ &\beta(\textstyle\sum_{h=1}^{c} p_{1,l_j+ic+h} + p)/(\beta+\alpha).
\end{aligned}$$

Then, we have

$$\begin{aligned}
C_j(\pi) &\leq (\textstyle\sum_{i=1}^{l_j} p_{1,i} + 2p - \tau) + \tfrac{\beta}{\beta+\alpha} \textstyle\sum_{i=0}^{k_j-1} (\textstyle\sum_{h=1}^{c} p_{1,l_j+ic+h} + p) \\
&= \tfrac{\alpha}{\beta+\alpha}(\textstyle\sum_{i=1}^{l_j} p_{1,i} + 2p - \tau) + \tfrac{\beta}{\beta+\alpha}\big(\textstyle\sum_{i=0}^{k-1}(\textstyle\sum_{h=1}^{c} p_{1,l+ic+h} + p) + (\textstyle\sum_{i=1}^{l_j} p_{1i} + 2p - \tau)\big) \\
&\leq \tfrac{\alpha}{\beta+\alpha} \cdot 3\beta + \tfrac{\beta}{\beta+\alpha}\big(\textstyle\sum_{i=1}^{j} p_{1,i} + jp/c + 2p - \tau\big) \\
&= \tfrac{\beta}{\beta+\alpha}\big(\textstyle\sum_{i=1}^{j}(p_{1,i} + p/c) + 3\alpha + 2p - \tau\big),
\end{aligned}$$

where $\sum_{i=1}^{l_j} p_{1,i} + 2p - \tau \leq 3\beta$ follows from the definition of β. Consequently, we have

$$\sum_{j=1}^{n} w_j C_j(\pi) \leq \frac{\beta}{\beta+\alpha}\Big(\sum_{j=1}^{n}\sum_{i=1}^{j} w_j(p_{1,i} + p/c) + (3\alpha + 2p - \tau)\sum_{j=1}^{n} w_j\Big). \tag{21}$$

Now let π^* be an optimal schedule of instance $\mathcal{J}$, and for each $j \in \{1, 2, \ldots, n\}$, let $J_{[j]}$ denote the job that occupies the jth position on the vehicle in π^*. Moreover, we consider the classical scheduling problem $1||\sum w_jC_j$ on job instance $\mathcal{J}' = \{J'_1, J'_2, \ldots, J'_n\}$, where each job J'_j has a processing time $p_{1,j} + p/c$ and a weight w_j. We consider the two schedules $\sigma = (J'_1, J'_2, \ldots, J'_n)$ and $\sigma^* = (J'_{[1]}, J'_{[2]}, \ldots, J'_{[n]})$ of problem $1||\sum w_jC_j$ on job instance $\mathcal{J}'$. From Smith [11], the well-known WSPT rule solves problem $1||\sum w_jC_j$ optimally. Thus, from the relations in (13), σ is an optimal schedule of problem $1||\sum w_jC_j$ on job instance $\mathcal{J}'$. It follows that $\sum_{j=1}^{n} w_jC_j(\sigma) \le \sum_{j=1}^{n} w_jC_j(\sigma^*)$. Note that $\sum_{j=1}^{n} w_jC_j(\sigma) = \sum_{j=1}^{n}\sum_{i=1}^{j} w_j(p_{1,i} + p/c)$ and $\sum_{j=1}^{n} w_jC_j(\sigma^*) = \sum_{j=1}^{n}\sum_{i=1}^{j} w_{[j]}(p_{1,[i]} + p/c)$. Then, we have

$$\sum_{j=1}^{n}\sum_{i=1}^{j} w_j(p_{1,i} + p/c) \le \sum_{j=1}^{n}\sum_{i=1}^{j} w_{[j]}(p_{1,[i]} + p/c). \tag{22}$$

By applying the inequality in Equation (22) to Equations (18) and (21), we obtain

$$\sum w_jC_j(\pi) \le \frac{3\beta}{\beta+\alpha} \sum w_{[j]}C_{[j]}(\pi^*)$$

This completes the proof. □

4. Conclusions

We studied the coordination of transportation and batching scheduling with one single vehicle for minimizing the total weighted completion time of the jobs without considering the processing cost of the batching machine. For this problem, we showed a unary NP-hardness of at least 3 for each batch capacity and presented a polynomial-time 3-approximation algorithm when the batch capacity is at least 2.

Future research may consider to include the processing cost in the objective function. In particular, approximation behavior of the problem $(1,c)|\tau|\sum w_jC_j + \alpha(b)$ with $\alpha(b) = \lambda b$ being a linear function in b is worthy of study. Moreover, when the batch capacity is given by $c = 2$, the computational complexity of problem $(1,2)|\tau|\sum w_jC_j + \alpha(b)$ is still open. A polynomial-time approximation scheme for solving problem $(1,c)|\tau|\sum w_jC_j$ is also expected.

Author Contributions: Conceptualization, methodology, and writing—original manuscript: H.W.; project management, supervision, and writing—review: J.Y.; data curation and formal analysis: Y.G.

Funding: This research was funded by the National Natural Science Foundation of China under grant numbers 11671368, 11771406, and 11901539. The third author was also supported by the China Postdoctoral Science Foundation under grant number 2019M652555.

Acknowledgments: The authors would like to thank the Associate Editor and the anonymous referees for their constructive comments and helpful suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tang, L.X.; Gong, H. The coordination of transportation and batching scheduling. *Appl. Math. Mod.* **2009**, *33*, 3854–3862. [CrossRef]
2. Hall, N.G.; Potts, C.N. The coordination of scheduling and batch deliveries. *Ann. Oper. Res.* **2005**, *135*, 41–64. [CrossRef]
3. Chen, Z.L. Integrated production and outbound distribution scheduling: review and extensions. *Oper. Res.* **2010**, *58*, 130–148. [CrossRef]
4. Zhu, H.L.; Leus, R.; Zhou, H. New results on the coordination of transportation and batching scheduling. *Appl. Math. Mod.* **2016**, *40*, 4016–4022. [CrossRef]
5. Allahverdi, A. The third comprehensive survey on scheduling problems with setup times/costs. *Eur. J. Oper. Res.* **2015**, *246*, 345–378. [CrossRef]
6. Ciavotta, M.; Meloni, C.; Pranzo, M. Speeding up a Rollout algorithm for complex parallel machine scheduling. *Internat. J. Prod. Res.* **2016**, *54*, 4993–5009. [CrossRef]

7. Wei, H.J.; Yuan, J.J. Two-machine flow-shop scheduling with equal processing time on the second machine for minimizing total weighted completion time. *Oper. Res. Letters.* **2019**, *47*, 41–46. [CrossRef]
8. Choi, B.C.; Yoon, S.H.; Chung, S.J. Minimizing the total weighted completion time in a two-machine proportionate flow shop with different machine speeds. *Internat. J. Prod. Res.* **2006**, *44*, 715–728. [CrossRef]
9. Hoogeveen, J.A.; Kawaguchi, T. Minimizing total completion time in a two-machine flowshop: analysis of special cases. *Math. Oper. Res.* **1999**, *24*, 887–913. [CrossRef]
10. Garey, M.R.; Johnson, D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*; Freeman: San Francisco, CA, USA, 1979.
11. Smith, W.E. Various optimizers for single stage production. *Naval Res. Logist. Quart.* **1956**, *3*, 59–66. [CrossRef]