Research Article

Parallel Machine Scheduling with Batch Delivery to Two Customers

Xueling Zhong¹ and Dakui Jiang²

¹Department of Internet Finance and Information Engineering, Guangdong University of Finance, Guangzhou 510520, China

²College of Management and Economics, Tianjin University, Tianjin 300072, China

Correspondence should be addressed to Xueling Zhong; zhongxuel@hotmail.com

Received 1 May 2015; Revised 22 August 2015; Accepted 31 August 2015

Academic Editor: Sergio Teggi

Copyright © 2015 X. Zhong and D. Jiang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In some make-to-order supply chains, the manufacturer needs to process and deliver products for customers at different locations. To coordinate production and distribution operations at the detailed scheduling level, we study a parallel machine scheduling model with batch delivery to two customers by vehicle routing method. In this model, the supply chain consists of a processing facility with m parallel machines and two customers. A set of jobs containing n_1 jobs from customer 1 and n_2 jobs from customer 2 are first processed in the processing facility and then delivered to the customers directly without intermediate inventory. The problem is to find a joint schedule of production and distribution such that the tradeoff between maximum arrival time of the jobs and total distribution cost is minimized. The distribution cost of a delivery shipment consists of a fixed charge and a variable cost proportional to the total distance of the route taken by the shipment. We provide polynomial time heuristics with worst-case performance analysis for the problem. If $m = 2$ and $(n_1 - b)(n_2 - b) < 0$, we propose a heuristic with worst-case ratio bound of $3/2$, where b is the capacity of the delivery shipment. Otherwise, the worst-case ratio bound of the heuristic we propose is $2 - 2/(m + 1)$.

1. Introduction

To meet the soaring demands of electronic devices in recent years, manufacturers in China start building new factories to increase production capacities. Two different strategies are mainly adopted by these manufacturers, one is to build a new factory at the undeveloped land near current factory and the other is to place the new factory to a different region with lower labor cost. Take Foxconn Technology Group, the world's largest electronics contractor manufacturer, for example, it not only built a new factory at Guanlan Technology Park after running one at Yousong Industrial District in Shenzhen city of China but also has been building many other factories at different regions of China. Clearly, it can share resources easily by adopting the former strategy and reduces production cost by adopting the latter one. Meanwhile, as a nonstandard parts supplier of a manufacturer adopting the former strategy, it should not only offer parts to the current factory but also provide parts to the new-built factory. In such applications, very little inventory of finished parts

exists at any point of time since nonstandard parts are custom made and the supplier will not start production early before it receives orders from the manufacturer. Hence, the production and distribution operations of the supplier are linked immediately, and the close linkage between production and distribution necessitates coordinating production and distribution operations at the level of detailed scheduling.

In this paper, we consider a parallel machine scheduling problem with batch delivery to two customers faced by the nonstandard part supplier in the above-described supply chain, which can be described as follows. There is a manufacturer, who has a set of $m > 2$ identical parallel machines facility, $M = \{1, 2, \dots, m\}$. At time zero, the manufacturer receives a set of n jobs, $N = \{1, 2, \dots, n\}$, from two customers 1 or 2, which are located at different locations in an underlying transportation network. Among jobs in N , jobs in the subset $N_k \subseteq N$ are ordered by customer k , $k = 1, 2$. Let $n_k = |N_k|$ denote the number of jobs in N_k . It is easy to see that $N = N_1 \cup N_2$ and $n = n_1 + n_2$. Each job in N should be processed onto one of the m machines in manufacturer and

then delivered to its customer in batch without intermediate inventory. Associated with each job $j \in N$ has a processing time of p_j units. Job preemption is not allowed; that is, processing a job on a machine cannot be interrupted until it is finished. All jobs and machines are available at time 0. All the finished jobs ordered by customer k need to be delivered in batch to customer k by the vehicle, $k = 1, 2$. Suppose that there are enough homogeneous vehicles available so that each vehicle will be used once and each delivery shipment will be transported by a dedicated vehicle. All the vehicles are stationed at the processing facility at time 0 and must go back to the facility once they complete a shipment. Each vehicle can carry up to at most b jobs in one shipment. The transportation cost incurred by each batch consists of a fixed charge f and a variable cost dependent on the particular route taken by the vehicle. We use c_{0i} , c_{ij} , and c_{j0} , respectively, to denote the variable cost for traveling from the processing facility to customer $i \in \{1, 2\}$, from customer $i \in \{1, 2\}$ to customer $j \in \{1, 2\}$, and from customer $i \in \{1, 2\}$ to the processing facility. The corresponding delivery times are denoted as t_{0i} , t_{ij} , and t_{j0} , respectively. We assume that $t_{ii} = c_{ii} = 0$, $t_{ij} = t_{ji}$, $c_{ij} = c_{ji}$, $t_{0i} + t_{ij} \geq t_{j0}$, and $c_{0i} + c_{ij} \geq c_{j0}$, $i, j = 0, 1, 2$. A delivery vehicle can depart from the processing facility only when all the jobs to be delivered have completed processing. We use D_j to denote the time when job $j \in N$ arrives to its customer. Define $D_{\max} = \max\{D_j \mid j \in N\}$ as the maximum arrival time to the two customers of jobs in N . The problem is to find a joint production and distribution schedule such that the tradeoff between maximum delivery time of the jobs and total distribution cost, that is, $\alpha D_{\max} + (1 - \alpha)T$, is minimized, where T denotes the total distribution cost, $D_{\max}$ measures customer service level, and $0 \leq \alpha \leq 1$ is a given constant and represents the decision maker's relative preference on customer service level and total distribution cost. Such an objective function is also adopted by Chen and Vairaktarakis [1].

This problem is a variation of the integrated production-distribution scheduling models with batch delivery to multiple customers by vehicle routing method, which is always encountered in make-to-order or time-sensitive product supply chains. In these supply chains, finished jobs are often delivered to customers immediately or shortly after the production which lead to production and distribution operations that are intimately linked with no intermediate inventory. Even though the research on integrated production-distribution scheduling models is fairly recent, much excellent work have been done in the last decade. Vehicle routing method is an efficient way to serve multiple customers by a shipment. However, only few literature studied the models with vehicle routing method (e.g., [1–4], among others). Chen and Vairaktarakis [1] studied a more general case, in which there are $k \geq 2$ customers in the supply chain. They proposed a heuristic with worst case ratio bound of $2 - 1/m$, the time complexity of which is $O(b^k n^{2k} k^{k-1})$. Clearly, when $k = 2$ the heuristic also needs to take $O(b^2 n^4)$ time. However, if jobs are delivered to the customers individually and immediately, Woeginger [5] provided a heuristic with worst case ratio bound of $2 - 2/(m + 1)$, the time complexity

of which is merely $O(n \log n)$. Hence, there may exist a heuristic that could outperform the heuristic proposed by Chen and Vairaktarakis [1]. According to this observation, we are interested in designing a more effective and efficient algorithm by adopting the approach proposed by Woeginger [5].

The rest of the paper is organized as follows. In Section 2, we give some optimality properties of the problem and introduce some notations. Then, we study the problem with $\alpha = 1$ or $\alpha = 0$ in Section 3 and with $0 < \alpha < 1$ in Section 4. Finally, in Section 5, we conclude the paper.

2. Notations and Preliminary Results

In this section, we first summarize the notations we used in our model and introduce some new notations in the following list:

- M : a set of m identical parallel machines in manufacturer.
- N : a set of n jobs ordered by two customers.
- N_k : a subset of jobs ordered by customer k , $k = 1, 2$.
- n_k : the number of jobs in N_k , $k = 1, 2$.
- p_j : the processing time of job j , $j \in N$.
- f : the fixed cost of using a vehicle.
- c_{0i} : the variable cost of traveling processing facility to customer j , $j \in N$.
- c_{ij} : the variable cost of traveling from customer i to customer j , $i = 1, 2$ and $j = 1, 2$.
- t_{0i} : the time of traveling processing facility to customer i , $i = 1, 2$.
- t_{ij} : the time of traveling from customer i to customer j , $i = 1, 2$ and $j = 1, 2$.
- D_j : the arrival time of job j to its customer, $j \in N$.
- $D_{\max}(\pi)$: the maximum arrival time of the jobs in solution π .
- $T(\pi)$: the total distribution cost in solution π .
- α : the preference on customer service level and total distribution cost, $0 \leq \alpha \leq 1$.
- $F(\pi)$: the objective value of solution π , $F(\pi) = \alpha D_{\max}(\pi) + (1 - \alpha)T(\pi)$.
- $D_{\max}^*$: the optimal objective value of the problem with $\alpha = 1$.
- T^* : the optimal objective value of the problem with $\alpha = 0$.
- S^* : the optimal solution of our problem.

We then present some straightforward optimality properties for the problem. Proofs are omitted.

Lemma 1. *There exists an optimal solution for the problem in which*

- (1) *there is no idle time between the jobs processed on each machine in the processing facility,*

- (2) the departure time of each shipment is the completion time of the last job included in the shipment,
- (3) jobs that are processed on the same machine and delivered in the same shipment are processed consecutively on that machine.

Lemma 2. $D_{\max}^* \leq D_{\max}(S^*)$ and $T^* \leq T(S^*)$.

3. The Problem with $\alpha = 1$ or $\alpha = 0$

In this section, we study two special cases of the problem with $\alpha = 1$ or $\alpha = 0$.

In the problem with $\alpha = 1$, because no cost is considered, each job is delivered in a separate vehicle immediately after it completes processing. Then, the problem is reduced to minimize $D_{\max}$. If $t_{01} = t_{02} = t_{12} = 0$, this case is equivalent to the classical parallel machine maximum completion time scheduling problem $P_m \parallel C_{\max}$, which is known to be NP-hard [6]. Hence, this case is also NP-hard. Woeginger [5] studied a model as the same as ours except that each job $j \in N$ is associated with an individual delivery time and develop a heuristic with worst-case ratio bound of $2 - 2/(m+1)$. Denote such a heuristic as MLS in [5]. The running complexity time of heuristic MLS is $O(n \log n)$. By the major idea of heuristic MLS, we propose a heuristic with worst-case ratio bound of $2 - 2/(m+1)$ for the problem with $\alpha = 1$ or $\alpha = 0$. Denote this heuristic as MMLS (Modified MLS). The running time complexity of heuristic MMLS is $O(n \log n)$. Similar to heuristic MLS, in heuristic MMLS, let $W_m(j) = (m-1)p_j + mt_{0j}$ denote a weight of job j depending on its processing time and delivery time, $j \in N$. The value of $W_m(j)$ can be computed in advance for each $j \in N$. Also, we let L_i represent the total processing time of jobs assigned to machine i so far during the algorithm execution, $i \in M$. Heuristic MMLS can be described in detail as follows.

Heuristic MMLS

Step 1. Place all jobs in a list sorted by nonincreasing weight $W_m(j)$. Set $L_i = 0$, for $i \in M$.

Step 2. Select a machine i for which L_i is as small as possible (select i arbitrarily if there is a tie). Choose the first unscheduled job j of the list and assign it to machine i . Set $L_i = L_i + p_j$. Repeat Step 2 until all jobs are scheduled.

Step 3. For all $i \in M$, resort the jobs assigned to machine i in Step 2 by nonincreasing delivery time. Denote this solution by $S_{\alpha=1}$.

Clearly, our problem with $\alpha = 1$ is a special case of the problem studied by Woeginger [5] and can also be solved by heuristic MMLS. Then, we have the following.

Theorem 3. $F(S_{\alpha=1}) \leq (2 - 2/(m+1))D_{\max}^*$.

The problem with $\alpha = 0$ is to minimize the total distribution cost. Because $D_{\max}$ is not considered, each vehicle should deliver as many jobs as possible to minimize the

total distribution cost. Thus, the following results hold for the problem with $\alpha = 0$. Before showing these results, we especially point out that job j directly delivered by a vehicle indicates that job j is delivered to its corresponding customer directly, and the vehicle loading job j does not pass by the other customer. We let $y_k = n_k - b(\lceil n_k/b \rceil - 1)$, $k = 1, 2$. The useful properties are presented as follows.

Lemma 4. There exists an optimal solution for the problem with $\alpha = 0$ in which $b(\lceil n_k/b \rceil - 1)$ jobs in N_k are directly delivered by $\lceil n_k/b \rceil - 1$ vehicles, each of which contains b jobs, $k = 1, 2$.

Corollary 5. If $c_{12} \geq f + c_{01} + c_{02}$, $\lceil n_k/b \rceil$ vehicles are used in the optimal distribution schedule of the problem with $\alpha = 0$ to direct deliver all jobs in N_k , each of which contains b jobs, $k = 1, 2$.

Corollary 6. If $y_1 y_2 = 0$, $\lceil n_k/b \rceil$ vehicles are used in the optimal distribution schedule of the problem with $\alpha = 0$ to direct deliver all jobs in N_k , each of which contains b jobs, $k = 1, 2$.

Corollary 7. If $y_1 + y_2 > b$, $\lceil n_k/b \rceil$ vehicles are used in the optimal distribution schedule of the problem with $\alpha = 0$ to direct deliver all jobs in N_k , each of which contains b jobs, $k = 1, 2$.

Corollary 8. If $c_{12} < f + c_{01} + c_{02}$ and $y_1 y_2 \neq 0$ and $y_1 + y_2 \leq b$, in the optimal distribution schedule of the problem with $\alpha = 0$, $\lceil n_k/b \rceil - 1$ vehicles are used to direct deliver $b(\lceil n_k/b \rceil - 1)$ jobs in N_k , each of which contains b jobs, $k = 1, 2$, one vehicle is used to deliver the remaining $y_1 + y_2$ jobs, which goes to customer 2 via customer 1 and then directly returns to the processing facility.

Since the above results are straightforward, we omit the proofs. Obviously, it is easy to construct the following polynomial-time exact algorithm for the problem with $\alpha = 0$ by adopting these results.

Algorithm EA

Step 1. Assign all jobs in N to the m machines in arbitrary sequence.

Step 2. If $c_{12} \geq f + c_{01} + c_{02}$, schedule the job delivery such that it satisfies Corollary 5. Otherwise, if $y_1 y_2 = 0$, schedule the job delivery such that it satisfies Corollary 6; if $y_1 + y_2 > b$, schedule the job delivery such that it satisfies Corollary 7; if $y_1 y_2 \neq 0$ and $y_1 + y_2 \leq b$, schedule the job delivery such that it satisfies Corollary 8.

4. The Problem with $0 < \alpha < 1$

When $0 < \alpha < 1$, the problem is NP-hard. This is because the following case of the problem $f = c_{0i} = t_{0i} = c_{ij} = t_{ij} = 0$, for $i = 1, 2$ and $j = 1, 2$, is equivalent to the classical NP-Hard problem $P_m \parallel C_{\max}$. Associated with n_1 and n_2 we study a case with $n_k < b$ for $k = 1, 2$ and the other cases. We will develop a heuristic for the case with $n_k < b$ for $k = 1, 2$ in

Section 4.1 and develop another heuristic for the other cases in Section 4.2, respectively. Particularly notice that the other cases include two types: the case with $n_k \geq b$ for $k = 1, 2$ and the case with $(n_1 - b)(n_2 - b) \leq 0$.

4.1. The Case with $n_k < b$ for $k = 1, 2$. We first introduce some straightforward optimality properties for this case, which will be used in the sequel. Proofs are easy, and we leave them to the reader.

Lemma 9. *There exists an optimal solution for the case $n_k < b$, $k = 1, 2$, and either $c_{12} > f + c_{01} + c_{02}$, or $n_1 + n_2 > b$, in which all the jobs for customer k are delivered by a single vehicle, $k = 1, 2$.*

In the following, we present a polynomial-time heuristic, denoted as H1, for the case and analyze its worst-case performance. In heuristic H1, similar to heuristic MMLS, we also let $W_m(j) = (m - 1)p_j + mt_{0j}$ denote a weight of job j depending on its processing time and delivery time, $j \in N$, and let L_i represent the total processing time of jobs assigned to machine i so far during the algorithm execution, $i \in M$. The value of $W_m(j)$ can be computed in advance for each $j \in N$. Note also that we provide a Largest Processing Time first (LPT) rule for sorting the jobs. Heuristic H1 can be described in detail as follows.

Heuristic H1

Step 1. If $n_1 + n_2 \leq b$ and $c_{12} < f + c_{01} + c_{02}$, go to Step 2; otherwise, go to Step 5.

Step 2. Place all jobs in a list in LPT order. Set $L_i = 0$, for $i \in M$.

Step 3. Select a machine i for which L_i is as small as possible (select i arbitrarily if there is a tie). Choose the first unscheduled job j of the list and assign it to machine i . Set $L_i = L_i + p_j$. Repeat Step 3 until all jobs are scheduled.

Step 4. Deliver all jobs in N by a single vehicle. In order to complete the travel, two routes can be opted for the vehicle. The first one is that the vehicle goes to customer 1 via customer 2 and then direct returns to the processing facility and the second one is that the vehicle goes to customer 2 via customer 1 and then directly returns to the processing facility. Denote the resulting solution adopting the first route and the second route by S_{22} and S_{23} , respectively.

Step 5. Place all jobs in a list sorted by nonincreasing weight $W_m(j)$. Set $L_i = 0$, for $i \in M$.

Step 6. Select a machine i for which L_i is as small as possible (select i arbitrarily if there is a tie). Choose the first unscheduled job j of the list and assign it to machine i . Set $L_i = L_i + p_j$. Set $L_i = L_i + p_j$. Repeat Step 6 until all jobs are scheduled.

Step 7. For all $i \in M$, resort the jobs assigned to machine i in Step 6 by nonincreasing delivery time.

Step 8. For $k = 1, 2$, deliver all jobs in N_k by a single vehicle. Denote the resulting solution by S_{23} .

Step 9. Denote the final solution by S_2 . If $n_1 + n_2 \leq b$ and $c_{12} < f + c_{01} + c_{02}$, select one from the three solutions S_{21} , S_{22} , and S_{23} with the lowest objective value as the final solution. Otherwise, set S_{23} as the final solution.

In heuristic H1, most of the computing time is used for sorting jobs. Hence, the time complexity of the heuristic is $O(n \log n)$. Next, we analyze its worst-case performance.

Theorem 10. $F(S_2) \leq (2 - 2/(m + 1))F(S^*)$.

Proof. There are three cases associated with n_1 and n_2 .

Case (i). If $n_1 + n_2 > b$, then by Lemma 9, the distribution schedule of S_{23} is an optimal one. Meanwhile, $D_{\max}(S_{23}) \leq (2 - 2/(m + 1))D_{\max}(S^*)$. Therefore, we have $F(S_{23}) \leq (2 - 2/(m + 1))F(S^*)$.

Case (ii). If $n_1 + n_2 \leq b$ and $c_{12} > f + c_{01} + c_{02}$, then by Lemma 9, the distribution schedule of S_{23} is an optimal one. Due to $D_{\max}(S_{23}) \leq (2 - 2/(m + 1))D_{\max}(S^*)$, we thus have $F(S_{23}) \leq (2 - 2/(m + 1))F(S^*)$.

Case (iii). If $n_1 + n_2 \leq b$ and $c_{12} \leq f + c_{01} + c_{02}$, there are two subcases associated with the distribution schedule of S^* .

Subcase (i). If two vehicles are used to deliver jobs, one vehicle delivers all jobs in N_1 and the other one delivers all jobs in N_2 ; then $F(S_{23}) \leq (2 - 2/(m + 1))F(S^*)$.

Subcase (ii). If only one vehicle is used to deliver all jobs in N , then $T(S^*) = f + c_{01} + c_{12} + c_{02}$ and the optimal route of the vehicle is one of the routes adopted by S_{21} or S_{22} . Therefore, the problem is reduced to the classical scheduling problem $P_m \parallel C_{\max}$. It is well known that the worst-case performance bound of the heuristic for this problem by applying LPT rule is $4/3 - 1/3m$ [7]. Thus, in this subcase, $\min\{F(S_{21}), F(S_{22})\} \leq (4/3 - 1/3m)F(S^*)$.

In Step 9 of the heuristic, we select the best of the three solutions; hence $F(S_2) \leq (2 - 2/(m + 1))F(S^*)$. $\square$

4.2. The Other Cases. As mentioned before, except the case with $n_k < b$ for $k = 1, 2$ the other cases include the case with $n_k \geq b$ for $k = 1, 2$ and the case with $(n_1 - b)(n_2 - b) \leq 0$. We have discussed the case with $n_k < b$ for $k = 1, 2$ in Section 4.1. In this subsection, we develop below a heuristic, denoted as H2, for the case with $n_k \geq b$ for $k = 1, 2$ and the case with $(n_1 - b)(n_2 - b) \leq 0$. Similar to heuristic MMLS, we also let $W_m(j) = (m - 1)p_j + mt_{0j}$ denote a weight of job j depending on its processing time and delivery time, $j \in N$, and let L_i represent the total processing time of jobs assigned to machine i so far during the algorithm execution, $i \in M$. The value of $W_m(j)$ can be computed in advance for each $j \in N$. Heuristic H2 can be described in detail as follows.

Heuristic H2

Step 1. Place all jobs in a list sorted by nonincreasing weight $W_m(j)$. Set $L_i = 0$, for $i \in M$.

Step 2. Select a machine i for which L_i is as small as possible (select i arbitrarily if there is a tie). Choose the first unscheduled job j of the list and assign it to machine i . Set $L_i = L_i + p_j$. Repeat Step 2 until all jobs are scheduled.

Step 3. For all $i \in M$, resort the jobs assigned to machine i in Step 2 by nonincreasing delivery time.

Step 4. For $k = 1, 2$, use $\lceil n_k/b \rceil$ vehicles to deliver jobs in N_k , where the first vehicle delivers $n_k - (\lceil n_k/b \rceil - 1)b$ jobs and the remaining $(\lceil n_k/b \rceil - 1)$ vehicles deliver b jobs. Denote the resulting solution by S_{11} .

Step 5. For $k = 1, 2$, use $\lceil n_k/b \rceil - 1$ vehicles to deliver the last $n_k - y_k$ jobs in N_k with largest processing completion time, each of which contains b jobs, where $y_k = n_k - b(\lceil n_k/b \rceil - 1)$. Use one vehicle to deliver the remaining $y_1 + y_2$ jobs. Two routes are available for the vehicle: the former one is that the vehicle goes to customer 1 via customer 2 and then directly returns to the processing facility and the latter one is that the vehicle goes to customer 2 via customer 1 and then directly returns to the processing facility. Denote the resulting solution adopting the former route and the latter route by S_{12} and S_{13} , respectively.

Step 6. Denote the final solution by S_1 . Select one from the three solutions S_{11} , S_{12} , and S_{13} with the smallest objective value as the final solution.

In heuristic H2, most of the computing time is used for sorting jobs in Step 1. Hence, the time complexity of the heuristic is $O(n \log n)$. Next, we analyze its worst case performance.

Theorem 11. $D_{\max}(S_{11}) \leq (2 - 2/(m+1))D_{\max}(S^*)$.

Proof. It is clear that $D_{\max}(S_{11}) = D_{\max}(S_{\alpha=1})$ since they adopt the same production schedule and the maximum arrival time of the jobs is independent of how the delivery shipments are formed when each job is direct delivered to its customer with no vehicle routing. In S_{11} , let P_k be the maximum processing completion time of jobs in N_k , for $k = 1, 2$. Hence, $D_{\max}(S_{11}) = D_{\max}(S_{\alpha=1}) = \max\{P_k + t_{0k} \mid k = 1, 2\}$. By Theorem 3, $D_{\max}(S_{11}) \leq (2 - 2/(m+1))D_{\max}^*$. Moreover, by Lemma 2, $D_{\max}(S^*) \geq D_{\max}^*$. Thus we have $D_{\max}(S_{11}) \leq (2 - 2/(m+1))D_{\max}(S^*)$. $\square$

For the case with $n_k \geq b$ for $k = 1, 2$ and the case with $(n_1 - b)(n_2 - b) \leq 0$, we have the following result.

Theorem 12. If $n_k \geq b$ for $k = 1, 2$, $T(S_{11}) \leq (4/3)T(S^*)$; if $(n_1 - b)(n_2 - b) < 0$, $T(S_{11}) \leq (3/2)T(S^*)$.

Proof. In S_{11} , $T(S_{11}) = \sum_{k \in K} \lceil n_k/b \rceil (f + 2c_{0k})$. There are three cases associated with y_1 and y_2 .

Case (i). If $y_1 y_2 = 0$, then, by Corollary 6, $T^* = \sum_{k=1}^2 \lceil n_k/b \rceil (f + 2c_{0k})$. Hence, $T(S_{11}) = T^*$.

Case (ii). If $y_1 + y_2 > b$, then, by Corollary 7, $T^* = \sum_{k=1}^2 \lceil n_k/b \rceil (f + 2c_{0k})$. Hence, $T(S_{11}) = T^*$.

Case (iii). If $y_1 y_2 > 0$ and $y_1 + y_2 \leq b$, then, by Corollary 8, we have

$$T^* = \sum_{k=1}^2 \left(\left\lceil \frac{n_k}{b} \right\rceil - 1 \right) (f + 2c_{0k}) + \min \{f + c_{01} + c_{12} + c_{02}, 2f + 2c_{01} + 2c_{02}\}. \quad (1)$$

There are two subcases associated with $T(S_{11})$ and T^* .

Subcase (i). If $f + c_{01} + c_{12} + c_{02} \geq 2f + 2c_{01} + 2c_{02}$, we have $T(S_{11}) = T^*$.

Subcase (ii). If $f + c_{01} + c_{12} + c_{02} < 2f + 2c_{01} + 2c_{02}$, we have $T^* = \sum_{k=1}^2 (\lceil n_k/b \rceil - 1)(f + 2c_{0k}) + f + c_{01} + c_{12} + c_{02}$. Now consider $T(S_{11})$:

$$\begin{aligned} T(S_{11}) &= \sum_{k=1}^2 \left(\left\lceil \frac{n_k}{b} \right\rceil - 1 \right) (f + 2c_{0k}) + 2f + 2c_{01} \\ &\quad + 2c_{02} \\ &= \sum_{k=1}^2 \left(\left\lceil \frac{n_k}{b} \right\rceil - 2 \right) (f + 2c_{0k}) \\ &\quad + \frac{4}{3} (3f + 3c_{01} + 3c_{02}). \end{aligned} \quad (2)$$

When $n_k \geq b$ and $y_1 y_2 > 0$, we have $\lceil n_k/b \rceil > n_k/b > 1$, for $k = 1, 2$. This implies that $\lceil n_k/b \rceil \geq 2$, $k = 1, 2$. Thus,

$$\begin{aligned} T(S_{11}) &\leq \frac{4}{3} \sum_{k=1}^2 \left(\left\lceil \frac{n_k}{b} \right\rceil - 2 \right) (f + 2c_{0k}) \\ &\quad + \frac{4}{3} (3f + 3c_{01} + c_{12} + 3c_{02}) = \frac{4}{3} T^*. \end{aligned} \quad (3)$$

When $(n_1 - b)(n_2 - b) < 0$ and $y_1 y_2 > 0$, there exists a $\bar{k} \in \{1, 2\}$ that satisfies $\lceil n_{\bar{k}}/b \rceil \geq 1$ and $\lceil n_{3-\bar{k}}/b \rceil \geq 2$. Thus,

$$\begin{aligned} T(S_{11}) &\leq \frac{3}{2} \sum_{k=1}^2 \left(\left\lceil \frac{n_k}{b} \right\rceil - 1 \right) (f + 2c_{0k}) + 2f + 2c_{01} \\ &\quad + 2c_{02} - \frac{1}{2} (f + 2c_{0\bar{k}}) \\ &= \frac{3}{2} \sum_{k=1}^2 \left(\left\lceil \frac{n_k}{b} \right\rceil - 1 \right) (f + 2c_{0k}) + \frac{3}{2} f + c_{0\bar{k}} \\ &\quad + 2c_{0(3-\bar{k})} = \frac{3}{2} T^* - \frac{1}{2} (3c_{12} - c_{0(3-\bar{k})} + c_{0\bar{k}}) \\ &\leq \frac{3}{2} T^*. \end{aligned} \quad (4)$$

Therefore, in both cases, we can conclude that $T(S_{11}) \leq (4/3)T^*$ if $n_k \geq b$ for $k = 1, 2$ and $T(S_1) \leq (3/2)T^*$ if $(n_1 - b)(n_2 - b) < 0$. By Lemma 2, $T^* \leq T(S^*)$. Therefore, $T = (S_{11}) \leq (4/3)T(S^*)$ if $n_1 \geq b$ and $n_2 \geq b$ and $T(S_{11}) \leq (3/2)T^*$ if $(n_1 - b)(n_2 - b) < 0$. $\square$

Corollary 13. If $m = 2$ and $(n_1 - b)(n_2 - b) < 0$, $F(S_1) \leq 3/2F(S^*)$; otherwise, $F(S_1) \leq (2 - 2/(m + 1))F(S^*)$.

Proof. By Theorem 11, $D_{\max}(S_{11}) \leq (2 - 2/(m + 1))D_{\max}(S^*)$. If $n_1 \geq b$ and $n_2 \geq b$, by Theorem 12, $T(S_{11}) \leq (4/3)T(S^*)$. Due to $4/3 \leq 2 - 2/(m + 1)$ when $m \geq 2$, we have $F(S_{11}) \leq (2 - 2/(m + 1))F(S^*)$. Otherwise, if $(n_1 - b)(n_2 - b) < 0$, by Theorem 12, $T(S_{11}) \leq (3/2)T(S^*)$. Due to $2 - 2/(m + 1) \geq 3/2$ when $m > 2$, we have $F(S_{11}) \leq (2 - 2/(m + 1))F(S^*)$. However, if $m = 2$, then $2 - 2/(m + 1) = 4/3 < 3/2$; if $m > 2$, then $2 - 2/(m + 1) \geq 3/2$. Hence, if $m = 2$ and $(n_1 - b)(n_2 - b) < 0$, $F(S_{11}) \leq (3/2)F(S^*)$; if $m > 2$ and $(n_1 - b)(n_2 - b) < 0$, $F(S_{11}) \leq (2 - 2/(m + 1))F(S^*)$. Because S_1 is one of the solutions S_{11}, S_{12} , and S_{13} with the smallest objective value, $F(S_1) \leq F(S_{11})$. This implies that $F(S_1) \leq 3/2F(S^*)$ if $m = 2$ and $(n_1 - b)(n_2 - b) < 0$, and $F(S_1) \leq (2 - 2/(m + 1))F(S^*)$ otherwise. $\square$

5. Conclusion

In this paper, we have studied a parallel machine scheduling model with batch delivery to two customers. The objective is to minimize the tradeoff between the maximum arrival time of the jobs and total distribution cost. Computational complexity of various cases of the problem has been clarified, and algorithms for these cases are provided. More specifically, we first provided a polynomial time heuristic with worst-case ratio bound of $2 - 2/(m + 1)$ and a polynomial time exact algorithm for the problem with $\alpha = 1$ and $\alpha = 0$, respectively; then we developed heuristics for the problem with $0 < \alpha < 1$. When the number of machines m is equal to 2 and the number of customer 1's orders n_1 and the number of customer 2's orders n_2 meet $(n_1 - b)(n_2 - b) < 0$, the worst-case ratio bound of our heuristic is $3/2$. Otherwise, the worst-case ratio of our developed heuristic is bounded by $2 - 2/(m + 1)$. The heuristic we proposed has the advantages over the existing heuristic proposed by Chen and Vairaktarakis [1] that it possesses tighter worst-case ratio bound and lower time complexity. However, in this paper we just consider the case with two customers in the supply chain; for future research it will be worth designing heuristics with tighter bound for multiple customers.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors thank the AE and the anonymous referees for their helpful comments and suggestions. This research was supported in part by the National Natural Science

Foundation of China (71501051), the Humanities and Social Sciences Research Foundation of Ministry of Education of China (13YJC630239), and the Foundation for Distinguished Young Teachers in Higher Education of Guangdong Province (YQ201403).

References

- [1] Z.-L. Chen and G. L. Vairaktarakis, "Integrated scheduling of production and distribution operations," *Management Science*, vol. 51, no. 4, pp. 614–628, 2005.
- [2] Y.-C. Chang and C.-Y. Lee, "Machine scheduling with job delivery coordination," *European Journal of Operational Research*, vol. 158, no. 2, pp. 470–487, 2004.
- [3] C.-L. Li and G. Vairaktarakis, "Coordinating production and distribution of jobs with bundling operations," *IEEE Transactions*, vol. 39, no. 2, pp. 203–215, 2007.
- [4] C. A. Ullrich, "Integrated machine scheduling and vehicle routing with time windows," *European Journal of Operational Research*, vol. 227, no. 1, pp. 152–165, 2013.
- [5] G. J. Woeginger, "Heuristics for parallel machine scheduling with delivery times," *Acta Informatica*, vol. 31, no. 6, pp. 503–512, 1994.
- [6] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, Calif, USA, 1979.
- [7] M. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, Springer, New York, NY, USA, 2012.