PARALLEL-MACHINE SCHEDULING IN SHARED MANUFACTURING

MIN JI, XINNA YE AND FANGYAO QIAN
School of Management and E-Business
Contemporary Business and Trade Research Center
Zhejiang Gongshang University
Hangzhou 310018, Zhejiang, P. R. China

T.C.E. CHENG
Department of Logistics and Maritime Studies
The Hong Kong Polytechnic University
Kowloon, Hong Kong

YIWEI JIANG∗
School of Management and E-Business
Contemporary Business and Trade Research Center
Zhejiang Gongshang University
Hangzhou 310018, Zhejiang, P. R. China

(Communicated by Ada Che)

ABSTRACT. We consider parallel-machine scheduling in the context of shared manufacturing where each job has a machine set to which it can be assigned for processing. Such a set is called the processing set. In the shared manufacturing setting, a job can be assigned not only to certain machines for processing, but can also be processed on the remaining machines at a certain cost. Compared with traditional scheduling with job rejection, the scheduling model under study embraces the notion of sustainable manufacturing. Showing that the problem is \( NP \)-hard, we develop a fully polynomial-time approximation scheme to solve the problem when the number of machines is fixed.

1. Introduction. In many applications of the parallel-machine scheduling model, the machines may have the same speed of processing jobs but different functions. Consequently, in such a manufacturing environment, each job has a restricted set of machines on which it can be processed, called its processing set. The general scheduling model with the processing set is as follows: There are \( n \) jobs \( \mathcal{J} = \{J_1, J_2, ..., J_n\} \) and \( m \) machines \( \mathcal{M} = \{M_1, M_2, ..., M_m\} \), whereby each job \( J_j \) is associated with

2020 Mathematics Subject Classification. 90B35, 90C27.
Key words and phrases. Scheduling, parallel machines, processing set, shared manufacturing, FPTAS.

This research was supported in part by the National Natural Science Foundation of China under grant numbers 11971434 and 11871327, Zhejiang Provincial Natural Science Foundation of China under grant number LY21G010002, and the Contemporary Business and Trade Research Center of Zhejiang Gongshang University, which is a key Research Institute of Social Sciences and Humanities of the Ministry of Education of China. Cheng was supported in part by The Hong Kong Polytechnic University under the Fung Yiu King - Wing Hang Bank Endowed Professorship in Business Administration.

∗ Corresponding author: Yiwei Jiang.
There are several significant special forms of the processing set, namely the nested processing set, the inclusive processing set, the interval processing set and the tree-hierarchical processing set. In the case of the nested processing set, for each machine pair \( M_j \) and \( M_k \), we have \( M_j \cap M_k = \emptyset \), \( M_j \subseteq M_k \), or \( M_k \subseteq M_j \). In the case of the inclusive processing set, for each machine pair \( M_j \) and \( M_k \), we have either \( M_j \subseteq M_k \) or \( M_k \subseteq M_j \). The special property of the interval processing set is that for any job \( J_{j} \), \( M_j = \{ M_{a_j}, M_{a_j+1}, \ldots, M_{b_j} \} \) for some \( 1 \leq a_j \leq b_j \leq m \). And for the tree-hierarchical processing set, associated with each machine set \( M_j \) is a node of a tree.

In the last decade, scheduling with processing set restrictions has received considerable research attention. Leung and Li [29] presented a survey of research on traditional parallel-machine scheduling with processing set restrictions. Since then, many researchers have pursued this line of research [11, 16]. Leung and Li [30] provided an expository literature review of this field. Li [33] considered parallel batching machine scheduling with the inclusive processing set restriction and non-identical machine capacity. Leung and Ng [31] provided fast approximation algorithms for uniform-machine scheduling with two kinds of processing sets.

Over the past 20 years, researchers have devoted much effort to study machine scheduling with rejection (MSR). Slotnick [41] and Shabtay et al. [40] provided comprehensive literature reviews on this topic. Ou et al. [35] studied parallel-machine scheduling that combines the inclusive processing set restriction and job rejection. They developed a \((\frac{2}{3} + \varepsilon)\)-approximation algorithm that runs in \( O \left( \frac{nm}{\varepsilon^2} \log P \right) \) time to minimize the makespan of all the accepted jobs plus the total rejection cost.

However, rejection may render resources under-utilized [26], which will lead to less social benefit. Therefore, we consider scheduling in the sharing environment. In the academic literature, there is no exact definition of the sharing notion. Price [38] suggested that sharing is the distribution of economic goods and services without considering the return in an intimate social group. Benkler [8] regarded sharing as a non-reciprocal pro-social behaviour. Belk [6] considered sharing as the act and process of distributing personal property to others for their use. Chasin et al. [10] held that the definition of sharing varies in different contexts where sharing takes place.

Applications of sharing can be found in many aspects of the modern life. Barnes and Mattsson [4] indicated that the sharing phenomenon and practice are expanding rapidly with the development of information and communications technology (ICT). Sharing knowledge [42], resources [43], experience [34], services [36], and products [22] are significant forms of the sharing practice. Many firms that practise sharing have achieved commercial success, see, e.g., Uber [27] and Airbnb [3].

Sharing has been a popular buzz word in academic circles. Belk [7] systematically analyzed the differences between sharing, gift giving, and commodity exchange. Lamberton and Rose [28] researched commercial sharing systems. Porter and Kramer [37] emphasized that sharing not only creates a bigger individual value but also leads to greater economic and social values. Albinsson and Perera [2] examined sharing events, and explored the connection between sharing, community building, and alternative consumption. Heinrichs [15] and Hamari et al. [13] revealed that sharing provides new possibilities for sustainable development.
et al. [17] proposed a new interpretation of shared value and illustrated the relationship between individual and shared values. Ryu et al. [39] conducted a survey of the sharing paradigm. Sharing has gradually penetrated into manufacturing. Brandt [9] first put forward the notion of shared manufacturing. Adhau et al. [1] revealed that manufacturing enterprises attempt to share manufacturing resources with other enterprises, in order to make better use of the resources and obtain greater benefits. Becker and Stern [5] studied resource sharing in transport logistics. They suggested that resource sharing in the manufacturing industry can contribute to the productivity and profitability of the collaborating enterprises. Li et al. [32] studied scheduling of shared distributed manufacturing resources. He and Zhang [14] systematically reviewed the advantages and research status of shared manufacturing. Jiang and Li [23] regarded shared manufacturing as a realized form of social manufacturing. They examined various kinds of shared factories that share production orders, processing resources, and manufacturing capability.

To the best of our knowledge, there is no research on scheduling in the context of shared manufacturing discussed above. Besides, as sharing contributes to sustainable development, it is reasonable to believe that scheduling in shared manufacturing can lead to enhancement of social benefits. Specifically, we consider parallel-machine scheduling that takes processing set restrictions into consideration in the shared manufacturing setting, seeking to develop an algorithm that can handle all the possibilities of processing set restrictions.

We organize the rest of the paper as follows: In Section 2 we introduce the problem under study. In Section 3 we provide a fully polynomial-time approximation scheme (FPTAS) for the problem. Finally, in Section 4, we conclude the paper and suggest topics for future research.

2. Problem description. We formally introduce the problem under study as follows: There are a set of \( n \) independent jobs \( J = \{ J_1, J_2, ..., J_n \} \) to be processed on a set of \( m \) identical parallel machines \( M = \{ M_1, M_2, ..., M_m \} \), whereby each job \( J_j \). Each machine can process at most one job at a time and job preemption is not allowed. All the jobs are available for processing at time 0. Each job \( J_j \) has a machine set \( M_j \) to which \( J_j \) can be assigned for processing without incurring any cost except the time cost, where \( M_j \) is an arbitrary subset of \( M \). If job \( J_j \) is assigned to one of machines in \( M_j \), the processing time \( p_{ij} \) is independent of the machines. However, if job \( J_j \) is assigned to machine \( M_i \) for processing, where \( M_i \in M\backslash M_j \), job \( J_j \) has the processing time \( p_{ij} \) and incurs an additional shared service cost \( w_{ij} \). Our goal is to find an optimal schedule to minimize the sum of the makespan and total shared service cost. We use the three-field notation [12] to denote the problem under study as \( P_m | M_j, SM | C_{\text{max}} + S \), where \( SM \) and \( S \) denote shared manufacturing and the total shared service cost, respectively.

We first show that the problem is \( NP \)-hard. It is easy to see that \( P_m | M_j | C_{\text{max}} \) is a special case of our problem when \( w_{ij} \rightarrow \infty, \forall i, j \). In addition, our scheduling problem reduces to problem \( P_m \| C_{\text{max}} \) when \( M_j = M_i \) for \( j = 1, 2, ..., n \). Since both problems \( P_m | M_j | C_{\text{max}} \) and \( P_m \| C_{\text{max}} \) are \( NP \)-hard, \( P_m | M_j, SM | C_{\text{max}} + S \) is \( NP \)-hard, too. In the following we provide an FPTAS to deal with this intractable problem.

3. FPTAS. An algorithm \( A \) is called a \((1+\varepsilon)\)-approximation algorithm for a minimization problem if it produces a solution that is at most \((1+\varepsilon)\) times as big as
the optimal value, running in time that is polynomial in the input size. A family of approximation algorithms \( \{A_\varepsilon\} \) is a fully polynomial-time approximation scheme if, for each \( \varepsilon > 0 \), algorithm \( A_\varepsilon \) is a \((1+\varepsilon)\)-approximation algorithm that is polynomial in the input size and in \( 1/\varepsilon \). From now on we assume, without loss of generality, that \( 0 < \varepsilon \leq 1 \). If \( \varepsilon > 1 \), then a 2-approximation algorithm can be taken as a \((1+\varepsilon)\)-approximation algorithm. In this section we propose an FPTAS for the problem under study based on the technique of trimming the state space (see, e.g., Kovalyov and Kubiak [24, 25]).

First, we introduce the variables \( x_j, j = 1, 2, \ldots, n \), where \( x_j = k \) if job \( J_j \) is assigned to machine \( M_k \), \( k \in \{1, 2, \ldots, m\} \), for processing. Let \( X \) denote the set of all the vectors \( x = (x_1, x_2, \ldots, x_n) \) with \( x_j = k, j = 1, 2, \ldots, n, k = 1, 2, \ldots, m \). We define the following initial and recursive functions on \( X \):

\[
\begin{align*}
  f^0_i(x) &= 0, \quad \text{for } i = 1, 2, \ldots, m, \\
  h_0(x) &= 0; \\
  \text{if } M_k \in M_j, \text{ then } \\
  f^k_j(x) &= f^k_{j-1}(x) + p_j, \\
  f^i_j(x) &= f^i_{j-1}(x), \quad \text{for } i \neq k, \\
  h_j(x) &= h_{j-1}(x); \\
  \text{else } \\
  f^k_j(x) &= f^k_{j-1}(x) + p_{kj}, \\
  f^i_j(x) &= f^i_{j-1}(x), \quad \text{for } i \neq k, \\
  h_j(x) &= h_{j-1}(x) + w_{kj}.
\end{align*}
\]

In addition,

\[
Q(x) = \max_{i=1, 2, \ldots, m} \{f^i_n(x)\} + h_n(x)
\]

Consequently, problem \( P_m | M_j, SM | C_{\max} + S \) reduces to the following problem:

Minimize \( Q(x) \) for \( x \in X \).

To solve the above problem, we introduce the procedure \( \text{Partition}(A, e, \delta) \) proposed by Kovalyov and Kubiak [24], where \( A \subseteq X \), \( e \) is a non-negative integer function on \( X \), and \( 0 < \delta \leq 1 \). The procedure partitions \( A \) into disjoint subsets \( A^1, A^2, \ldots, A^e \) such that \( |e(x) - e(x')| \leq \delta \min \{e(x), e(x')\} \) for any \( x, x' \in A^e \), \( h = 1, 2, \ldots, k_e \). We provide the details of \( \text{Partition}(A, e, \delta) \) in the following.

**Procedure \( \text{Partition}(A, e, \delta) \)**

**Step 1:** \( x \in A \) in the order \( x^{(1)}, x^{(2)}, \ldots, x^{(|A|)} \) such that \( 0 \leq e(x^{(1)}) \leq e(x^{(2)}) \leq \cdots \leq e(x^{(|A|)}) \).

**Step 2:** Assign the vectors \( x^{(1)}, x^{(2)}, \ldots, x^{(i_1)} \) to set \( A^1 \) until \( i_1 \) is found such that \( e(x^{(i_1)}) \leq (1 + \delta) e(x^{(1)}) \) and \( e(x^{(i_1+1)}) > (1 + \delta) e(x^{(1)}) \). If such \( i_1 \) does not exist, then take \( A^1_k = A^1 \) and stop.

Assign the vectors \( x^{(i_1+1)}, x^{(i_1+2)}, \ldots, x^{(i_2)} \) to set \( A^2 \) until \( i_2 \) is found such that \( e(x^{(i_2)}) \leq (1 + \delta) e(x^{(i_1+1)}) \) and \( e(x^{(i_2+1)}) > (1 + \delta) e(x^{(i_1+1)}) \). If such \( i_2 \) does not exist, then take \( A^2_k = A^2 = A - A^1 \) and stop.

Continue the above construction until \( x^{(|A|)} \) is included in \( A^e_k \) for some \( k_e \).
Procedure Partition\((A, e, \delta)\) requires \(O(|A| \log |A|)\) operations to arrange the vectors of \(A\) in non-decreasing order of \(e(v)\) and \(O(|A|)\) operations to provide a partition. We present two main properties of Partition that we will use in the development of our FPTAS \(A_e\). Kovalyov and Kubiak [24, 25] first provided the properties, which Ji and Cheng [18, 19], and Ji et al. [20, 21] extended subsequently.

**Property 1.** \(|e(x) - e(x')| \leq \delta \min \{e(x), e(x')\}\) for any \(x, x' \in A^*_j\), \(j = 1, 2, \ldots, k_e\).

**Property 2.** \(k_e \leq \log c(x^{|A|})/\delta + 2\) for \(0 < \delta \leq 1\) and \(1 \leq e(x^{|A|})\).

We provide a formal description of the FPTAS \(A_e\) for problem \(P_m |M_j, SM| C_{\text{max}} + S\) as follows:

**Algorithm \(A_e\)**

**Step 1.** (Initialization) Set \(Y_0 = \{(0, 0, \ldots, 0)\}\) and \(j = 1\).

**Step 2.** (Generation of \(Y_1, Y_2, \ldots, Y_n\)) For set \(Y_{j-1}\), generate \(Y'_j\) by adding \(k, k = 1, 2, \ldots, m\), in position \(j\) of each vector from \(Y_{j-1}\). Calculate the following for any \(x \in Y_j\), assuming \(x_j = k\):

\[
\begin{align*}
\text{if } M_k & \in M_j, \text{ then } \\
 f_j^k(x) & = f_{j-1}^k(x) + p_j, \\
 f_j^i(x) & = f_{j-1}^i(x), \text{ for } i \neq k, \\
 h_j(x) & = h_{j-1}(x); \\
\text{else } \\
 f_j^k(x) & = f_{j-1}^k(x) + p_k, \\
 f_j^i(x) & = f_{j-1}^i(x), \text{ for } i \neq k, \\
 h_j(x) & = h_{j-1}(x) + w_k.
\end{align*}
\]

If \(j = n\), then set \(Y_n = Y'_n\) and go to Step 3.

If \(j < n\), then set \(\delta = \varepsilon/2(n + 1)\) and perform the following computation.

Invoke Partition \((Y'_j, f'_j, \delta) (i = 1, 2, \ldots, m)\) to partition set \(Y'_j\) into disjoint sets \(Y'_1, Y'_2, \ldots, Y'_k, \ldots, Y'_k, \ldots, Y'_k\).

Invoke Partition \((Y'_j, h_j, \delta)\) to partition set \(Y'_j\) into disjoint sets \(Y^h, Y^h, \ldots, Y^h\).

Divide set \(Y'_j\) into the following subsets \(Y_{a_1 \ldots a_m} = Y^h \cap \ldots \cap Y_{a_m}^h \cap Y^h, a_1 = 1, 2, \ldots, k; a_m = 1, 2, \ldots, k; b = 1, 2, \ldots, k_h\). For each non-empty subset \(Y_{a_1 \ldots a_m}\), choose a vector \(x^{|x^{|x^{|x^{|x|a_1 \ldots a_m}b|a_1 \ldots a_m}b|a_1 \ldots a_m}b|a_1 \ldots a_m}b|\) such that \(\min_{x \in Y_{a_1 \ldots a_m}} \{f_j(x) + h_j(x)\}\).

**Step 3.** (Solution) Select vector \(x^0 \in Y_n\) such that \(Q(x^0) = \min \{Q(x) | x \in Y_n\} = \min \{\max_{i=1,2,\ldots,m} \{f_i(x) + h_i(x) \} | x \in Y_n\}\).

Let \(x^* = (x^*_1, x^*_2, \ldots, x^*_n)\) be an optimal solution for \(P_m |M_j, SM| C_{\text{max}} + S\) and \(L = \log (\max \{n, p_{\text{max}}, w_{\text{max}}\})\), where

\[
p_{\text{max}} = \max_{j=1,2,\ldots,n} \{p_j, \max \{p_k | M_k \in M_j \}\}
\]

and \(w_{\text{max}} = \max_{j=1,2,\ldots,n} \{w_k | M_k \in M_j \}\).
Theorem 3.1. Algorithm $A_z$ finds $x^0 \in X$ for problem $P_m |M_j, SM| C_{max} + S$ such that $Q(x^0) \leq (1 + \varepsilon)Q(x^*)$ in $O(n^{m+2}L^{m+2}/\varepsilon^{m+1})$ time.

Proof. Suppose that $(x_1^*, x_2^*, \ldots, x_j^*, 0, \ldots, 0) \in Y_{a_1 \ldots a_m} \subseteq Y_j'$ for some $j$ and $a_1 \ldots a_m$. By the definition of algorithm $A_z$, such $j$ always exists, e.g., $j = 1$. Algorithm $A_z$ may not choose $(x_1^*, x_2^*, \ldots, x_j^*, 0, \ldots, 0)$ for further construction; however, a vector $x^{(a_1 \ldots a_m)}$ is chosen instead. By Property 1, we have

$$|f_i^j(x^*) - f_i^j(x^{(a_1 \ldots a_m)})| \leq \delta f_i^j(x^*), \quad i = 1, 2, \ldots, m,$$

and

$$|h_j(x^*) - h_j(x^{(a_1 \ldots a_m)})| \leq \delta h_j(x^*).$$

Set $\delta_1 = \delta$. We consider vector $(x_1^*, x_2^*, x_j^*, 0, \ldots, 0)$ and

$$\tilde{x}^{(a_1 \ldots a_m)} = (x_1^{a_1 \ldots a_m}, \ldots, x_j^{a_1 \ldots a_m}, x_{j+1}^*, 0 \ldots 0).$$

Without loss of generality, we assume that $x_{j+1}^* = k$.

For $M_k \in M_j$, we have

$$|f_{j+1}^k(x^*) - f_{j+1}^k(\tilde{x}^{(a_1 \ldots a_m)})| = |f_{j+1}^k(x^*) + p_j| - (f_{j+1}^k(\tilde{x}^{(a_1 \ldots a_m)}) + p_j) = |f_{j+1}^k(x^*) - f_{j+1}^k(\tilde{x}^{(a_1 \ldots a_m)})| \leq \delta_1 f_{j+1}^k(x^*) \leq \delta_1 f_{j+1}^k(x^*).$$

For $M_k \in M \setminus M_j$, we have

$$|f_{j+1}^k(x^*) - f_{j+1}^k(\tilde{x}^{(a_1 \ldots a_m)})| = |(f_{j+1}^k(x^*) + p_{kj})| - (f_{j+1}^k(\tilde{x}^{(a_1 \ldots a_m)}) + p_{kj}) = |f_{j+1}^k(x^*) - f_{j+1}^k(\tilde{x}^{(a_1 \ldots a_m)})| \leq \delta_1 f_{j+1}^k(x^*) \leq \delta_1 f_{j+1}^k(x^*).$$

As a consequence, we obtain

$$|f_{j+1}^k(x^*) - f_{j+1}^k(\tilde{x}^{(a_1 \ldots a_m)})| \leq \delta_1 f_{j+1}^k(x^*). \quad (1)$$

Similarly, for $i$, we have

$$|f_{j+1}^i(x^*) - f_{j+1}^i(\tilde{x}^{(a_1 \ldots a_m)})| \leq \delta_1 f_{j+1}^i(x^*). \quad (2)$$

Therefore, combining (1) and (2), we conclude that

$$|f_{j+1}^i(x^*) - f_{j+1}^i(\tilde{x}^{(a_1 \ldots a_m)})| \leq \delta_1 f_{j+1}^i(x^*), \quad \text{for } i = 1, 2, \ldots, m. \quad (3)$$

Furthermore, for $M_k \in M_j$, we have

$$|h_{j+1}(x^*) - h_{j+1}(\tilde{x}^{(a_1 \ldots a_m)})| = |h_j(x^*) - h_j(\tilde{x}^{(a_1 \ldots a_m)})| \leq \delta_1 h_j(x^*) \leq \delta_1 h_{j+1}(x^*).$$
For \( M_k \in \mathcal{M} \setminus \mathcal{M}_j \), we have
\[
\left| h_{j+1}(x^*) - h_{j+1}(\tilde{x}^{(a_1 \ldots a_m b)}) \right| \\
= \left| (h_j(x^*) + w_k) - (\tilde{h}_{j+1}(\tilde{x}^{(a_1 \ldots a_m b)}) + w_k) \right| \\
= \left| h_j(x^*) - h_j(\tilde{x}^{(a_1 \ldots a_m b)}) \right| \leq \delta_1 h_j(x^*) \leq \delta_1 h_{j+1}(x^*).
\]
Consequently, we obtain
\[
\left| h_{j+1}(x^*) - h_{j+1}(\tilde{x}^{(a_1 \ldots a_m b)}) \right| \leq \delta_1 h_{j+1}(x^*). \tag{4}
\]

Thus, considering (3) and (4), we conclude that
\[
f_{j+1}^i\left(\tilde{x}^{(a_1 \ldots a_m b)}\right) \leq (1 + \delta_1) f_{j+1}^i(x^*), \quad \text{for } i = 1, 2, \ldots, m \tag{5}
\]
and
\[
h_{j+1}\left(\tilde{x}^{(a_1 \ldots a_m b)}\right) \leq (1 + \delta_1) h_{j+1}(x^*). \tag{6}
\]

Assume that \( \tilde{x}^{(a_1 \ldots a_m b)} \in Y_{c_1 \ldots c_m d} \subseteq Y'_{j-1} \) and algorithm \( \mathcal{A}_c \) chooses \( x^{(c_1 \ldots c_m d)} \) instead of \( \tilde{x}^{(a_1 \ldots a_m b)} \) in the \( j+1 \) iteration. It follows that
\[
\left| f_{j+1}^i\left(\tilde{x}^{(a_1 \ldots a_m b)}\right) - f_{j+1}^i\left(x^{(c_1 \ldots c_m d)}\right)\right| \\
\leq \delta f_{j+1}^i\left(\tilde{x}^{(a_1 \ldots a_m b)}\right) \leq \delta (1 + \delta_1) f_{j+1}^i(x^*), \quad i = 1, 2, \ldots, m \tag{7}
\]
and
\[
\left| h_{j+1}\left(\tilde{x}^{(a_1 \ldots a_m b)}\right) - h_{j+1}\left(x^{(c_1 \ldots c_m d)}\right)\right| \\
\leq \delta h_{j+1}\left(\tilde{x}^{(a_1 \ldots a_m b)}\right) \leq \delta (1 + \delta) h_{j+1}(x^*). \tag{8}
\]

By (5) and (7), we have
\[
\left| f_{j+1}^i(x^*) - f_{j+1}^i(x^{(c_1 \ldots c_m d)})\right| \\
\leq \left| f_{j+1}^i(x^*) - f_{j+1}^i\left(\tilde{x}^{(a_1 \ldots a_m b)}\right)\right| + \left| f_{j+1}^i\left(\tilde{x}^{(a_1 \ldots a_m b)}\right) - f_{j+1}^i(x^{(c_1 \ldots c_m d)})\right| \tag{9}
\]
\[
\leq (\delta_1 + \delta (1 + \delta_1)) f_{j+1}^i(x^*) \\
\leq (\delta + \delta_1 (1 + \delta)) f_{j+1}^i(x^*).
\]

Similarly, by (6) and (8), we conclude that
\[
\left| h_{j+1}(x^*) - h_{j+1}(x^{(c_1 \ldots c_m d)})\right| \\
\leq \left| h_{j+1}(x^*) - h_{j+1}\left(\tilde{x}^{(a_1 \ldots a_m b)}\right)\right| + \left| h_{j+1}\left(\tilde{x}^{(a_1 \ldots a_m b)}\right) - h_{j+1}\left(x^{(c_1 \ldots c_m d)}\right)\right| \tag{10}
\]
\[
\leq (\delta_1 + \delta (1 + \delta_1)) h_{j+1}(x^*) \leq (\delta + \delta_1 (1 + \delta)) h_{j+1}(x^*).
\]

Set \( \delta_l = \delta + \delta_{l-1} (1 + \delta), \quad l = 2, 3, \ldots, n - j + 1 \). From (9) and (10), we conclude that
\[
f_{j+1}^i\left(x^{(c_1 \ldots c_m d)}\right) \leq \delta_2 f_{j+1}^i(x^*)
\]
and
\[
h_{j+1}\left(x^{(c_1 \ldots c_m d)}\right) \leq \delta_2 h_{j+1}(x^*).
Repeating the above argument for \( j + 2, \ldots, n \), we show that there exists \( x' \in Y_n \) such that
\[
f_n^i (x^{(c_1 \ldots c_m d)}) \leq \delta_{n-j+1} f_n^i (x^*) , \quad \text{for } i = 1, 2, \ldots, m
\]
and
\[
h_n (x^{(c_1 \ldots c_m d)}) \leq \delta_{n-j+1} h_n (x^*) .
\]
Thus, we obtain
\[
\delta_{n-j+1} \leq \delta \sum_{j=0}^{n} (1 + \delta)^j = (1 + \delta)^{n+1} - 1
\]
\[
= \sum_{j=1}^{n+1} \frac{(n+1) \cdot (n-j+2)}{j!} \delta^j = \sum_{j=1}^{n+1} \frac{(n+1) \cdot (n-j+2)}{j!(n+1)^j} \left( \frac{\varepsilon}{2} \right)^j
\]
\[
\leq \sum_{j=1}^{n+1} \frac{1}{j!} \left( \frac{\varepsilon}{2} \right)^j \leq \sum_{j=1}^{n+1} \left( \frac{\varepsilon}{2} \right)^j \leq \varepsilon \sum_{j=1}^{n+1} \left( \frac{1}{2} \right)^j \leq \varepsilon
\]
It follows that
\[
f_n^i (x') \leq (1 + \varepsilon) f_n^i (x^*) , \quad \text{for } i = 1, 2, \ldots, m ,\quad (11)
\]
and
\[
h_n (x') \leq (1 + \varepsilon) h_n (x^*) .\quad (12)
\]
By (11) and (12), we have
\[
Q (x') = \max_{i=1,2,\ldots,m} \left\{ f_n^i (x') \right\} + h_n (x') = \delta f_n^k (x') + h_n (x') \leq (1+\varepsilon) \left( f_n^k (x^*) + h_n (x^*) \right)
\]
\[
\leq (1+\varepsilon) \left( \max_{i=1,2,\ldots,m} \left\{ f_n^i (x^*) \right\} + h_n (x^*) \right) = (1+\varepsilon) Q (x^*)
\]
Thus, in Step 3 of algorithm \( \mathcal{A}_\varepsilon \), vector \( x^0 \) will be chosen such that
\[
Q (x^0) \leq Q (x') \leq (1 + \varepsilon) Q (x^*) .
\]
To analyze the time complexity of algorithm \( \mathcal{A}_\varepsilon \), we see that the most time-consuming operation in iteration of Step 2 is a call of procedure \( \text{Partition} \), which requires \( O \left( |Y'_j| \log |Y'_j| \right) \) time to execute. To estimate \( |Y'_j| \), recall that \( |Y'_{j+1}| \leq m |Y'_j| \leq mk_{f_j} \ldots k_{f_{n-1}} k_h \). By Property 2, we have
\[
k_{f_j} \leq 2(n+1) \log (np_{\max})/\varepsilon + 2
\]
\[
\leq 4(n+1) \log \max \{n,p_{\max}\}/\varepsilon + 2 \leq 4(n+1)L/\varepsilon + 2 , \quad \text{for } i = 1, 2, \ldots, m
\]
and
\[
k_h \leq 2(n+1) \log (nw_{\max})/\varepsilon + 2
\]
\[
\leq 4(n+1) \log \max \{n,w_{\max}\}/\varepsilon + 2 \leq 4(n+1)L/\varepsilon + 2 .
\]
Therefore, we get \( |Y'_j| = O \left( n^{m+1} L^{m+1}/\varepsilon^{m+1} \right) \) and
\[
|Y'_j| \log |Y'_j| = O \left( n^{m+1} L^{m+2}/\varepsilon^{m+1} \right) .
\]
As a consequence, the time complexity of algorithm \( \mathcal{A}_\varepsilon \) is \( O \left( n^{m+2} L^{m+2}/\varepsilon^{m+1} \right) .\) \( \square \)
4. Conclusions. We consider parallel-machine scheduling with processing set restrictions in the shared manufacturing setting. Different from traditional scheduling with job rejection, the jobs in our problem cannot be rejected. Under the scheduling model in shared manufacturing, a job can be assigned to certain machines for processing, or it can be processed on the remaining machines at a certain cost. Since resource sharing is conductive to sustainable development, it is reasonable to believe that the scheduling model under study embraces the notion of sustainable manufacturing. We give an FPTAS to solve the problem in which the number of machines is fixed. Future research may explore other scheduling models with various objectives in the context of shared manufacturing.

Acknowledgments. We are thankful to two anonymous referees for their helpful comments and suggestions on an earlier version of our paper.

REFERENCES

[1] S. Adhau, M. L. Mittal and A. Mittal, A multi-agent system for decentralized multi-project scheduling with resource transfers, International Journal of Production Economics, 146 (2013), 646–661.
[2] P. A. Albinsson and B. Y. Perera, Alternative marketplaces in the 21st century: Building community through sharing events, Journal of Consumer Behaviour, 11 (2012), 303–315.
[3] S. Amaro, L. Andreu and S. Huang, Millennials’ intentions to book on Airbnb, Current Issues in Tourism, 22 (2019), 2284–2298.
[4] S. J. Barnes and J. Mattsson, Building tribal communities in the collaborative economy: An innovation framework, Prometheus, 34 (2016), 95–113.
[5] T. Becker and H. Stern, Impact of resource sharing in manufacturing on logistical key figures, Procedia CIRP, 41 (2016), 579–584.
[6] R. Belk, Why not share rather than own?, Annals of the American Academy of Political and Social Science, 611 (2007), 126–140.
[7] R. Belk, Sharing, Journal of Consumer Research, 36 (2010), 715–734.
[8] Y. Benkler, Sharing nicely: On shareable goods and the emergence of sharing as a modality of economic production, The Yale Law Journal, 114 (2004), 273–358.
[9] E. Brandt, A vision for shared manufacturing, Mechanical Engineering, 112 (1990), 52–55.
[10] F. Chasin, M. von Hoffen, M. Cramer and M. Matzner, Peer-to-peer sharing and collaborative consumption platforms: A taxonomy and a reproducible analysis, Information Systems and E-Business Management, 16 (2018), 293–325.
[11] L. Epstein and A. Levin, Scheduling with processing set restrictions: PTAS results for several variants, International Journal of Production Economics, 133 (2011), 586–595.
[12] R. L. Graham, E. L. Lawler, J. K. Lenstra and A. H. G. Rinnooy Kan, Optimization and approximation in deterministic sequencing and scheduling: A survey, Annals of Discrete Mathematics, 5 (1979), 287–326.
[13] J. Hamari, M. Sjöklint and A. Ukkonen, The sharing economy: Why people participate in collaborative consumption, Journal of the Association for Information Science and Technology, 67 (2016), 2047–2059.
[14] J. He, J. Zhang and X. Gu, Research on sharing manufacturing in Chinese manufacturing industry, International Journal of Advanced Manufacturing Technology, 104 (2019), 463–476.
[15] H. Heinrichs, Sharing economy: A potential new pathway to sustainability, Gaia-ecological Perspectives for Science and Society, 22 (2013), 228–231.
[16] Y. Huo and J. Y. T. Leung, Parallel machine scheduling with nested processing set restrictions, European Journal of Operational Research, 204 (2010), 229–236.
[17] K. N. Irvine, L. O’Brien, N. Ravenscroft, N. Cooper, M. Everard, I. Fazey, M. S. Reed and J. O. Kenter, Ecosystem services and the idea of shared values, Ecosystem Services, 21 (2016), 184–193.
[18] M. Ji and T. C. E. Cheng, Parallel-machine scheduling with simple linear deterioration to minimize total completion time, European Journal of Operational Research, 188 (2008), 342–347.
[19] M. Ji and T. C. E. Cheng, Batch scheduling of simple linear deteriorating jobs on a single machine to minimize makespan, European Journal of Operational Research, 202 (2010), 90–98.
[20] M. Ji, X. Tang, X. Zhang and T. C. E. Cheng, Machine scheduling with deteriorating jobs and DeJong’s learning effect, Computers and Industrial Engineering, 91 (2016), 42–47.
[21] M. Ji, Q. Yang, D. Yao and T. C. E. Cheng, Single-machine batch scheduling of linear deteriorating jobs, Theoretical Computer Science, 580 (2015), 36–49.
[22] B. Jiang and L. Tian, Collaborative consumption: Strategic and economic implications of product sharing, Management Science, 64 (2018), 1171–1188.
[23] P. Jiang and P. Li, Shared factory: A new production node for social manufacturing in the context of sharing economy, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 234 (2020), 285–294.
[24] M. Y. Kovalyov and W. Kubiak, A fully polynomial approximation scheme for minimizing makespan of deteriorating jobs, Journal of Heuristics, 4 (1998), 287–297.
[25] M. Y. Kovalyov and W. Kubiak, A fully polynomial approximation scheme for the weighted earliness-tardiness problem, Operations Research, 47 (1999), 757–761.
[26] H. Kurdi, E. Aloboud, S. Alhassan and E. T. Alotaibi, An algorithm for handling starvation and resource rejection in public clouds, Procedia Computer Science, 34 (2014), 242–248.
[27] Z. W. Y. Lee, T. K. H. Chan, M. S. Balaji and A. Y.-L. Cheng, Why people participate in the sharing economy: An empirical investigation of Uber, Internet Research, 28 (2018), 829–850.
[28] C. P. Lamberton and R. L. Rose, When is ours better than mine? A framework for understanding and altering participation in commercial sharing systems, Social Science Electronic Publishing, 76 (2011), 109–125.
[29] J. Y.-T. Leung and C.-L. Li, Scheduling with processing set restrictions: A survey, International Journal of Production Economics, 116 (2008), 251–262.
[30] J. Y.-T. Leung and C.-L. Li, Scheduling with processing set restrictions: A literature update, International Journal of Production Economics, 175 (2016), 1–11.
[31] J. Y.-T. Leung and C. T. Ng, Fast approximation algorithms for uniform machine scheduling with processing set restrictions, European Journal of Operational Research, 260 (2017), 507–513.
[32] K. Li, T. Zhou, B. Liu and H. Li, A multi-agent system for sharing distributed manufacturing resources, Expert Systems with Applications, 99 (2018), 32–43.
[33] S. Li, Parallel batch scheduling with inclusive processing set restrictions and non-identical capacities to minimize makespan, European Journal of Operational Research, 260 (2017), 12–20.
[34] A. M. Munar and J. K. S. Jacobsen, Motivations for sharing tourism experiences through social media, Tourism Management, 43 (2014), 46–54.
[35] J. Ou, X. Zhong and X. Qi, Scheduling parallel machines with inclusive processing set restrictions and job rejection, Naval Research Logistics, 63 (2016), 667–681.
[36] A. Paagman, M. Tate, E. Furtmueller and J. de Bloom, An integrative literature review and empirical validation of motives for introducing shared services in government organizations, International Journal of Information Management, 35 (2015), 110–123.
[37] M. E. Porter and M. R. Kramer, Creating shared value, Harvard Business Review, Harvard Business Review, 89 (2011), 62–77.
[38] J. A. Price, Sharing: The integration of intimate economies, Anthropologica (New Series), 17 (1975), 3–27.
[39] H. Ryu, M. Basu and O. Saito, What and how are we sharing? A systematic review of the sharing paradigm and practices, Sustainability Science, 14 (2019), 515–527.
[40] D. Shabtay, N. Gaspar and M. Kaspi, A survey on offline scheduling with rejection, Journal of Scheduling, 16 (2013), 3–28.
[41] S. A. Slotnick, Order acceptance and scheduling: A taxonomy and review, European Journal of Operational Research, 212 (2011), 1–11.
[42] S. Wang and R. A. Noe, Knowledge sharing: A review and directions for future research, Human Resource Management Review, 20 (2010), 115–131.
[43] C. H. Yu, K. Doppler, C. B. Ribeiro and O. Tirkkonen, Resource sharing optimization for device-to-device communication underlaying cellular networks, IEEE Transactions on Wireless Communications, 10 (2011), 2752–2763.

Received May 2020; revised August 2020.
E-mail address: jimkeen@163.com
E-mail address: Edwin.Cheng@polyu.edu.hk
E-mail address: ywjiang@zjgsu.edu.cn