# Approximation Algorithms and a Hardness Result for the Three-Machine Proportionate Mixed Shop

Longcheng Liu[1,4], Guanqun Ni[2,4], Yong Chen[3,4], Randy Goebel[4], Yue Luo[1], An Zhang[3,4], and Guohui Lin[4](✉)

[1] School of Mathematical Sciences, Xiamen University, Xiamen, China
longchengliu@xmu.edu.cn
[2] College of Management, Fujian Agriculture and Forestry University, Fuzhou, China
guanqunni@163.com
[3] Department of Mathematics, Hangzhou Dianzi University, Hangzhou, China
{chenyong,anzhang}@hdu.edu.cn
[4] Department of Computing Science, University of Alberta, Edmonton, AB, Canada
{rgoebel,guohui}@ualberta.ca

**Abstract.** A mixed shop is to process a mixture of a set of flow-shop jobs and a set of open-shop jobs. Mixed shops are in general much harder than flow-shops and open-shops, and have been studied since the 1980's. We consider the three machine proportionate mixed shop problem denoted as $M3 \mid prpt \mid C_{\max}$, in which each job has equal processing times on all three machines. Koulamas and Kyparisis (Eur J Oper Res 243:70–74, 2015) showed that the problem is solvable in polynomial time in some very special cases; for the non-solvable case, they proposed a 5/3-approximation algorithm. In this paper, we present an improved 4/3-approximation algorithm and show that this ratio of 4/3 is asymptotically tight; when the largest job is a flow-shop job, we present a fully polynomial-time approximation scheme (FPTAS). On the negative side, while the $F3 \mid prpt \mid C_{\max}$ problem is polynomial-time solvable, we show an interesting hardness result that adding one open-shop job to the job set makes the problem NP-hard if this open-shop job is larger than any flow-shop job.

**Keywords:** Scheduling · Mixed shop · Proportionate
Approximation algorithm
Fully polynomial-time approximation scheme

## 1 Introduction

We study in this paper the following three-machine proportionate mixed shop, denoted as $M3 \mid prpt \mid C_{\max}$ in the three-field notation [4]. Given three machines $M_1, M_2, M_3$ and a set $\mathcal{J} = \mathcal{F} \cup \mathcal{O}$ of jobs, where $\mathcal{F} = \{J_1, J_2, \ldots, J_\ell\}$ and $\mathcal{O} = \{J_{\ell+1}, J_{\ell+2}, \ldots, J_n\}$, each job $J_i \in \mathcal{F}$ needs to be processed non-preemptively


S. Tang et al. (Eds.): AAIM 2018, LNCS 11343, pp. 268–280, 2018.
https://doi.org/10.1007/978-3-030-04618-7_22

through $M_1, M_2, M_3$ sequentially with a processing time $p_i$ on each machine and each job $J_i \in \mathcal{O}$ needs to be processed non-preemptively on $M_1, M_2, M_3$ in any machine order with a processing time $q_i$ on each machine. The scheduling constraint is usual in that at every time point a job can be processed by at most one machine and a machine can process at most one job. The objective is to minimize the maximum job completion time, *i.e.*, the makespan.

The jobs of $\mathcal{F}$ are referred to as *flow-shop jobs* and the jobs of $\mathcal{O}$ are called *open-shop jobs*. The mixed shop is to process such a mixture of a set of flow-shop jobs and a set of open-shop jobs. We assume without loss of generality that $p_1 \geq p_2 \geq \ldots \geq p_\ell$ and $q_{\ell+1} \geq q_{\ell+2} \geq \ldots \geq q_n$.

Mixed shops have many real-life applications and have been studied since the 1980's. The scheduling of medical tests in an outpatient health care facility and the scheduling of classes/exams in an academic institution are two typical examples, where the patients (students, respectively) must complete a number of medical tests (academic activities, respectively); some of these activities must be done in the same sequential order while the others can be finished in any order; and the time-spans for all these activities should not overlap with each other. The *proportionate* shops were also introduced in the 1980's [11] and they are one of the most specialized shops with respect to the job processing times which have received many studies [12].

Masuda et al. [10] and Strusevich [16] considered the two-machine mixed shop problem to minimize the makespan, *i.e.*, $M2 \mid\mid C_{\max}$; they both showed that the problem is polynomial time solvable. Shakhlevich and Sotskov [14] studied mixed shops for processing two jobs with an arbitrary regular objective function. Brucker [1] surveyed the known results on the mixed shop problems either with two machines or for processing two jobs. Shakhlevich et al. [13] studied the mixed shop problems with more than two machines for processing more than two jobs, with or without preemption. Shakhlevich et al. [15] reviewed the complexity results on the mixed shop problems with three or more machines for processing a constant number of jobs.

When $\mathcal{O} = \emptyset$, the $M3 \mid prpt \mid C_{\max}$ problem reduces to the $F3 \mid prpt \mid C_{\max}$ problem, which is solvable in polynomial time [2]. When $\mathcal{F} = \emptyset$, the problem reduces to the $O3 \mid prpt \mid C_{\max}$ problem, which is ordinary (or called weakly) NP-hard [8]. It follows that the $M3 \mid prpt \mid C_{\max}$ problem is at least ordinary NP-hard. Recently, Koulamas and Kyparisis [7] showed that for some very special cases, the $M3 \mid prpt \mid C_{\max}$ problem is solvable in polynomial time; for the non-solvable case, they showed an absolute performance bound of $2\max\{p_1, q_{\ell+1}\}$ and presented a 5/3-approximation algorithm.

In this paper, we design an improved 4/3-approximation algorithm for (the non-solvable case of) the $M3 \mid prpt \mid C_{\max}$ problem, and show that the performance ratio of 4/3 is asymptotically tight. When the largest job is a flow-shop job, that is $p_1 \geq q_{\ell+1}$, we present a *fully polynomial-time approximation scheme* (FPTAS). On the negative side, while the $F3 \mid prpt \mid C_{\max}$ problem is polynomial-time solvable, we show an interesting hardness result that adding one single open-shop job to the job set makes the problem NP-hard if this open-shop job is larger than any flow-shop job. We construct the reduction from the well-known Partition problem [3].

The rest of the paper is organized as follows. In Sect. 2, we introduce some notations and present a lower bound on the optimal makespan $C^*_{\max}$. We present in Sect. 3 the FPTAS for the $M3 \mid prpt \mid C_{\max}$ problem when $p_1 \geq q_{\ell+1}$. The 4/3-approximation algorithm for the case where $p_1 < q_{\ell+1}$ is presented in Sect. 4, and the performance ratio of 4/3 is shown to be asymptotically tight. We show in Sect. 5 that, when there is only one open-shop job $J_n$ and $p_1 < q_n$, the $M3 \mid prpt \mid C_{\max}$ problem is NP-hard, through a reduction from the Partition problem. We conclude the paper with some remarks in Sect. 6.

## 2 Preliminaries

For any subset of jobs $\mathcal{X} \subseteq \mathcal{F}$, the *total processing time* of the jobs of $\mathcal{X}$ on one machine is denoted as

$$P(\mathcal{X}) = \sum_{J_i \in \mathcal{X}} p_i.$$

For any subset of jobs $\mathcal{Y} \subseteq \mathcal{O}$, the *total processing time* of the jobs of $\mathcal{Y}$ on one machine is denoted as

$$Q(\mathcal{Y}) = \sum_{J_i \in \mathcal{Y}} q_i.$$

The set minus operation $\mathcal{J} \setminus \{J\}$ for a single job $J \in \mathcal{J}$ is abbreviated as $\mathcal{J} \setminus J$ throughout the paper.

Given that the *load* (*i.e.*, the total job processing time) of each machine is $P(\mathcal{F}) + Q(\mathcal{O})$, the job $J_{\ell+1}$ has to be processed by all three machines, and one needs to process all the flow-shop jobs of $\mathcal{F}$, the following lower bound on the optimum $C^*_{\max}$ is established [2,7]:

$$C^*_{\max} \geq \max\{P(\mathcal{F}) + Q(\mathcal{O}),\ 3q_{\ell+1},\ 2p_1 + P(\mathcal{F})\}. \quad (1)$$

## 3 An FPTAS for the Case Where $p_1 \geq q_{\ell+1}$

In this section, we design an approximation algorithm $A(\epsilon)$ for the $M3 \mid prpt \mid C_{\max}$ problem when $p_1 \geq q_{\ell+1}$, for any given $\epsilon > 0$. The algorithm $A(\epsilon)$ produces a schedule $\pi$ with its makespan $C^{\pi}_{\max} < (1+\epsilon)C^*_{\max}$, and its running time is polynomial in both $n$ and $1/\epsilon$.

Consider a bipartition $\{\mathcal{A}, \mathcal{B}\}$ of the job set $\mathcal{O} = \{J_{\ell+1}, J_{\ell+2}, \ldots, J_n\}$, *i.e.*, $\mathcal{A} \cup \mathcal{B} = \mathcal{O}$ and $\mathcal{A} \cap \mathcal{B} = \emptyset$. Throughout the paper, a part of the bipartition is allowed to be empty. The following *procedure* Proc$(\mathcal{A}, \mathcal{B}, \mathcal{F})$ produces a schedule $\pi$:

1. the jobs of $\mathcal{F}$ are processed in the *longest processing time* (LPT) order on all three machines, and every job is processed first on $M_1$, then on $M_2$, lastly on $M_3$;
2. the jobs of $\mathcal{A}$ are processed in the LPT order on all three machines, and every one is processed first on $M_2$, then on $M_3$, lastly on $M_1$;

3. the jobs of $\mathcal{B}$ are processed in the LPT order on all three machines, and every one is processed first on $M_3$, then on $M_1$, lastly on $M_2$; and
4. the machine $M_1$ processes (the jobs of) $\mathcal{F}$ first, then $\mathcal{B}$, lastly $\mathcal{A}$, denoted as $\langle \mathcal{F}, \mathcal{B}, \mathcal{A} \rangle$;
5. the machine $M_2$ processes $\mathcal{A}$ first, then $\mathcal{F}$, lastly $\mathcal{B}$, denoted as $\langle \mathcal{A}, \mathcal{F}, \mathcal{B} \rangle$;
6. the machine $M_3$ processes $\mathcal{B}$ first, then $\mathcal{A}$, lastly $\mathcal{F}$, denoted as $\langle \mathcal{B}, \mathcal{A}, \mathcal{F} \rangle$.

Proc$(\mathcal{A}, \mathcal{B}, \mathcal{F})$ runs in $O(n \log n)$ time to produce the schedule $\pi$, of which an illustration is shown in Fig. 1.

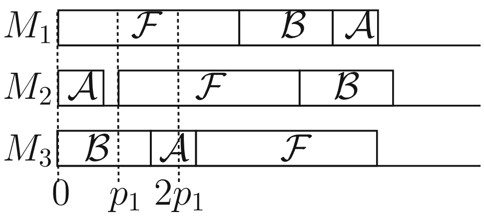


**Fig. 1.** An illustration of the schedule $\pi$ produced by Proc$(\mathcal{A}, \mathcal{B}, \mathcal{F})$, where $\{\mathcal{A}, \mathcal{B}\}$ is a bipartition of the set $\mathcal{O}$ and the jobs of each of $\mathcal{A}, \mathcal{B}, \mathcal{F}$ are processed in the LPT order on all three machines.

The following two lemmas state that if both $Q(\mathcal{A}) \le p_1$ and $Q(\mathcal{B}) \le p_1$, or both $Q(\mathcal{A}) \ge p_1$ and $Q(\mathcal{B}) \ge p_1$, then the schedule $\pi$ produced by Proc$(\mathcal{A}, \mathcal{B}, \mathcal{F})$ is optimal. Due to the space limit, we refer the readers to our arXiv submission [9] for the detailed proofs.

**Lemma 1** [9]**.** *If both $Q(\mathcal{A}) \le p_1$ and $Q(\mathcal{B}) \le p_1$, then the schedule $\pi$ produced by* Proc$(\mathcal{A}, \mathcal{B}, \mathcal{F})$ *is optimal, with its makespan $C^{\pi}_{\max} = C^*_{\max} = 2p_1 + P(\mathcal{F})$.*

**Lemma 2** [9]**.** *If both $Q(\mathcal{A}) \ge p_1$ and $Q(\mathcal{B}) \ge p_1$, then the schedule $\pi$ produced by* Proc$(\mathcal{A}, \mathcal{B}, \mathcal{F})$ *is optimal, with its makespan $C^{\pi}_{\max} = C^*_{\max} = P(\mathcal{F}) + Q(\mathcal{O})$.*

Now we are ready to present the approximation algorithm $A(\epsilon)$, for any $\epsilon > 0$.

In the first step, we check whether $Q(\mathcal{O}) \le p_1$ or not. If $Q(\mathcal{O}) \le p_1$, then we run Proc$(\mathcal{O}, \emptyset, \mathcal{F})$ to construct a schedule $\pi$ and terminate the algorithm. The schedule $\pi$ is optimal by Lemma 1.

In the second step, the algorithm $A(\epsilon)$ constructs an instance of the Knapsack problem [3], in which there is an item corresponding to the job $J_i \in \mathcal{O}$, also denoted as $J_i$. The item $J_i$ has a profit $q_i$ and a size $q_i$. The capacity of the knapsack is $p_1$. The Min-Knapsack problem is to find a subset of items of minimum profit that *cannot* be packed into the knapsack, and it admits an FPTAS [6]. The algorithm $A(\epsilon)$ runs a $(1 + \epsilon)$-approximation algorithm for the Min-Knapsack problem to obtain a job subset $\mathcal{A}$. It then runs Proc$(\mathcal{A}, \mathcal{O} \setminus \mathcal{A}, \mathcal{F})$ to construct a schedule, denoted as $\pi^1$.

The Max-Knapsack problem is to find a subset of items of maximum profit that can be packed into the knapsack, and it admits an FPTAS, too [5]. In the third step, the algorithm $A(\epsilon)$ runs a $(1-\epsilon)$-approximation algorithm for the Max-Knapsack problem to obtain a job subset $\mathcal{B}$. Then it runs $\text{Proc}(\mathcal{O} \setminus \mathcal{B}, \mathcal{B}, \mathcal{F})$ to construct a schedule, denoted as $\pi^2$.

The algorithm $A(\epsilon)$ outputs the schedule with a smaller makespan between $\pi^1$ and $\pi^2$. A high-level description of the algorithm $A(\epsilon)$ is provided in Fig. 2.

Algorithm $A(\epsilon)$:

1. If $Q(\mathcal{O}) \le p_1$, then run $\text{Proc}(\mathcal{O}, \emptyset, \mathcal{F})$ to produce a schedule $\pi$; output the schedule $\pi$.
2. Construct an instance of Knapsack, where an item $J_i$ corresponds to the job $J_i \in \mathcal{O}$; $J_i$ has a profit $q_i$ and a size $q_i$; the capacity of the knapsack is $p_1$.
   2.1. Run a $(1+\epsilon)$-approximation for Min-Knapsack to obtain a job subset $\mathcal{A}$.
   2.2. Run $\text{Proc}(\mathcal{A}, \mathcal{O} \setminus \mathcal{A}, \mathcal{F})$ to construct a schedule $\pi^1$.
3. 3.1. Run a $(1-\epsilon)$-approximation for Max-Knapsack to obtain a job subset $\mathcal{B}$.
   3.2. Run $\text{Proc}(\mathcal{O} \setminus \mathcal{B}, \mathcal{B}, \mathcal{F})$ to construct a schedule $\pi^2$.
4. Output the schedule with a smaller makespan between $\pi^1$ and $\pi^2$.

**Fig. 2.** A high-level description of the algorithm $A(\epsilon)$.

In the following performance analysis, we assume without of loss of generality that $Q(\mathcal{O}) > p_1$. We have the following (in-)equalities inside the algorithm $A(\epsilon)$:

$$\text{OPT}^1 = \min\{Q(\mathcal{X}) \mid \mathcal{X} \subseteq \mathcal{O},\ Q(\mathcal{X}) > p_1\}; \quad (2)$$

$$p_1 < Q(\mathcal{A}) \le (1+\epsilon)\text{OPT}^1; \quad (3)$$

$$\text{OPT}^2 = \max\{Q(\mathcal{Y}) \mid \mathcal{Y} \subseteq \mathcal{O},\ Q(\mathcal{Y}) \le p_1\}; \quad (4)$$

$$p_1 \ge Q(\mathcal{B}) \ge (1-\epsilon)\text{OPT}^2, \quad (5)$$

where $\text{OPT}^1$ ($\text{OPT}^2$, respectively) is the optimum to the constructed Min-Knapsack (Max-Knapsack, respectively) problem.

**Lemma 3.** *In the algorithm $A(\epsilon)$, if $Q(\mathcal{O} \setminus \mathcal{A}) \le p_1 - \epsilon\text{OPT}^1$, then for any bipartition $\{\mathcal{X}, \mathcal{Y}\}$ of the job set $\mathcal{O}$, $Q(\mathcal{X}) > p_1$ implies $Q(\mathcal{Y}) \le p_1$.*

*Proof.* Note that the job subset $\mathcal{A}$ is computed in Step 2.1 of the algorithm $A(\epsilon)$, and it satisfies Eq. (3). By the definition of $\text{OPT}^1$ in Eq. (2) and using Eq. (3), we have $Q(\mathcal{X}) \ge \text{OPT}^1 \ge Q(\mathcal{A}) - \epsilon\text{OPT}^1$. Furthermore, from the fact that $Q(\mathcal{O}) = Q(\mathcal{X}) + Q(\mathcal{Y}) = Q(\mathcal{A}) + Q(\mathcal{O} \setminus \mathcal{A})$ and the assumption that $Q(\mathcal{O} \setminus \mathcal{A}) \le p_1 - \epsilon\text{OPT}^1$, we have

$$\begin{aligned} Q(\mathcal{Y}) &= Q(\mathcal{A}) + Q(\mathcal{O} \setminus \mathcal{A}) - Q(\mathcal{X}) \\ &\le Q(\mathcal{A}) + Q(\mathcal{O} \setminus \mathcal{A}) - (Q(\mathcal{A}) - \epsilon \mathrm{OPT}^1) \\ &= Q(\mathcal{O} \setminus \mathcal{A}) + \epsilon \mathrm{OPT}^1 \\ &\le p_1 - \epsilon \mathrm{OPT}^1 + \epsilon \mathrm{OPT}^1 \\ &= p_1. \end{aligned}$$

This finishes the proof of the lemma. □

**Lemma 4.** *In the algorithm $A(\epsilon)$, if $Q(\mathcal{O} \setminus \mathcal{A}) \le p_1 - \epsilon \mathrm{OPT}^1$, then $C^*_{\max} \ge P(\mathcal{F}) + Q(\mathcal{O}) + p_1 - \mathrm{OPT}^2$.*

*Proof.* Consider an arbitrary optimal schedule $\pi^*$ that achieves the makespan $C^*_{\max}$. Note that the flow-shop job $J_1$ is first processed on the machine $M_1$, then on machine $M_2$, and last on machine $M_3$.

In the schedule $\pi^*$, let $S_i$ and $C_i$ be the start processing time and the finish processing time of the job $J_1$ on the machine $M_i$, respectively, for $i = 1, 2, 3$. On the machine $M_2$, let $\mathcal{J}^1 = \mathcal{O}^1 \cup \mathcal{F}^1$ denote the subset of jobs processed before $J_1$, and $\mathcal{J}^2 = \mathcal{O}^2 \cup \mathcal{F}^2$ denote the subset of jobs processed after $J_1$, where $\{\mathcal{O}^1, \mathcal{O}^2\}$ is a bipartition of the job set $\mathcal{O}$ and $\{\mathcal{F}^1, \mathcal{F}^2\}$ is a bipartition of the job set $\mathcal{F} \setminus J_1$. Also, let $\delta_1$ and $\delta_2$ denote the total amount of machine idle time for $M_2$ before processing $J_1$ and after processing $J_1$, respectively (see Fig. 3 for an illustration).

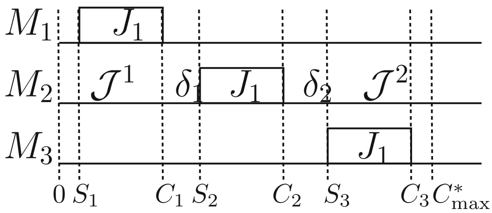


**Fig. 3.** An illustration of an optimal schedule $\pi^*$, in which $\mathcal{J}^1$ and $\mathcal{J}^2$ are the subsets of jobs processed on $M_2$ before $J_1$ and after $J_1$, respectively; $\delta_1$ and $\delta_2$ are the total amount of machine idle time for $M_2$ before processing $J_1$ and after processing $J_1$, respectively.

Note that $\mathcal{F} = J_1 \cup \mathcal{F}^1 \cup \mathcal{F}^2$ is the set of flow-shop jobs. The job $J_1$ and the jobs of $\mathcal{F}^1$ should be finished before time $S_2$ on the machine $M_1$, and the job $J_1$ and the jobs of $\mathcal{F}^2$ can only be started after time $C_2$ on the machine $M_3$. That is,

$$p_1 + P(\mathcal{F}^1) \le S_2 \tag{6}$$

and

$$p_1 + P(\mathcal{F}^2) \le C^*_{\max} - C_2. \tag{7}$$

If $Q(\mathcal{O}^1) \le p_1$, then we have $Q(\mathcal{O}^1) \le \mathrm{OPT}^2$ by the definition of $\mathrm{OPT}^2$ in Eq. (4). Combining this with Eq. (6), we achieve that $\delta_1 = S_2 - P(\mathcal{F}^1) - Q(\mathcal{O}^1) \ge p_1 - \mathrm{OPT}^2$.

If $Q(\mathcal{O}^1) > p_1$, then we have $Q(\mathcal{O}^2) \le p_1$ by Lemma 3. Hence, $Q(\mathcal{O}^2) \le \mathrm{OPT}^2$ by the definition of $\mathrm{OPT}^2$ in Eq. (4). Combining this with Eq. (7), we achieve that $\delta_2 = C^*_{\max} - C_2 - P(\mathcal{F}^2) - Q(\mathcal{O}^2) \ge p_1 - \mathrm{OPT}^2$.

The last two paragraphs prove that $\delta_1 + \delta_2 \ge p_1 - \mathrm{OPT}^2$. Therefore,

$$\begin{aligned} C^*_{\max} &= Q(\mathcal{O}^1) + P(\mathcal{F}^1) + \delta_1 + p_1 + Q(\mathcal{O}^2) + P(\mathcal{F}^2) + \delta_2 \\ &= P(\mathcal{F}) + Q(\mathcal{O}) + \delta_1 + \delta_2 \\ &\ge P(\mathcal{F}) + Q(\mathcal{O}) + p_1 - \mathrm{OPT}^2. \end{aligned}$$

This finishes the proof of the lemma. □

**Lemma 5.** *In the algorithm $A(\epsilon)$, if $Q(\mathcal{O} \setminus \mathcal{A}) \le p_1 - \epsilon \mathrm{OPT}^1$, then $C^{\pi^2}_{\max} < (1+\epsilon) C^*_{\max}$.*

*Proof.* Denote $\overline{\mathcal{B}} = \mathcal{O} \setminus \mathcal{B}$. Note that the job set $\mathcal{B}$ computed in Step 3.1 of the algorithm $A(\epsilon)$ satisfies $p_1 \ge Q(\mathcal{B}) \ge (1-\epsilon)\mathrm{OPT}^2$, and the schedule $\pi^2$ is constructed by $\textsc{Proc}(\overline{\mathcal{B}}, \mathcal{B}, \mathcal{F})$. We distinguish the following two cases according to the value of $Q(\overline{\mathcal{B}})$.

Case 1. $Q(\overline{\mathcal{B}}) \le p_1$. In this case, the schedule $\pi^2$ is optimal by Lemma 1.

Case 2. $Q(\overline{\mathcal{B}}) > p_1$. The schedule $\pi^2$ constructed by $\textsc{Proc}(\overline{\mathcal{B}}, \mathcal{B}, \mathcal{F})$ has the following properties (see Fig. 4 for an illustration):

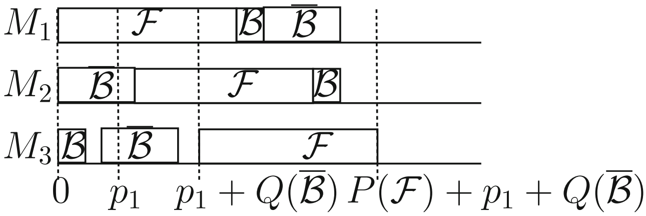


**Fig. 4.** An illustration of the schedule $\pi^2$ constructed by $\textsc{Proc}(\overline{\mathcal{B}}, \mathcal{B}, \mathcal{F})$ in Case 2, where $Q(\mathcal{B}) \le p_1$ and $Q(\overline{\mathcal{B}}) > p_1$. The machines $M_1$ and $M_2$ do not idle; the machine $M_3$ may idle between processing the job set $\mathcal{B}$ and the job set $\overline{\mathcal{B}}$ and may idle between processing the job set $\overline{\mathcal{B}}$ and the job set $\mathcal{F}$. $M_3$ starts processing the job set $\mathcal{F}$ at time $p_1 + Q(\overline{\mathcal{B}})$.

1. The jobs are processed consecutively on the machine $M_1$ since $J_1$ is the largest job. The completion time of $M_1$ is thus $C_1^{\pi^2} = Q(\mathcal{O}) + P(\mathcal{F})$.
2. The jobs are processed consecutively on the machine $M_2$ due to $Q(\mathcal{B}) \le p_1$ and $Q(\overline{\mathcal{B}}) > p_1$. The completion time of $M_2$ is thus $C_2^{\pi^2} = Q(\mathcal{O}) + P(\mathcal{F})$.
3. The machine $M_3$ starts processing the job set $\mathcal{F}$ consecutively at time $p_1 + Q(\overline{\mathcal{B}})$ due to $Q(\mathcal{B}) \le p_1$. The completion time of $M_3$ is $C_3^{\pi^2} = P(\mathcal{F}) + p_1 + Q(\overline{\mathcal{B}})$.

Note that $C_3^{\pi^2} = P(\mathcal{F}) + p_1 + Q(\overline{\mathcal{B}}) \geq P(\mathcal{F}) + Q(\mathcal{B}) + Q(\overline{\mathcal{B}}) = Q(\mathcal{O}) + P(\mathcal{F})$, implying $C_{\max}^{\pi^2} = P(\mathcal{F}) + p_1 + Q(\overline{\mathcal{B}})$. Combining Eq. (5) with Lemma 4, we have

$$\begin{aligned} C_{\max}^{\pi^2} &= P(\mathcal{F}) + p_1 + Q(\overline{\mathcal{B}}) \\ &= P(\mathcal{F}) + Q(\mathcal{O}) + p_1 - Q(\mathcal{B}) \\ &\leq P(\mathcal{F}) + Q(\mathcal{O}) + p_1 - (1-\epsilon)\mathrm{OPT}^2 \\ &\leq C_{\max}^* + \epsilon \mathrm{OPT}^2 \\ &< (1+\epsilon) C_{\max}^*, \end{aligned}$$

where the last inequality is due to $\mathrm{OPT}^2 \leq p_1 < C_{\max}^*$. This finishes the proof of the lemma. □

**Lemma 6.** *In the algorithm $A(\epsilon)$, if $p_1 - \epsilon\mathrm{OPT}^1 < Q(\mathcal{O} \setminus \mathcal{A}) < p_1$, then $C_{\max}^{\pi^1} < (1+\epsilon)C_{\max}^*$.*

*Proof.* Denote $\overline{\mathcal{A}} = \mathcal{O} \setminus \mathcal{A}$. Note that the job set $\mathcal{A}$ computed in Step 2.1 of the algorithm $A(\epsilon)$ satisfies $p_1 < Q(\mathcal{A}) \leq (1+\epsilon)\mathrm{OPT}^1$, and the schedule $\pi^1$ is constructed by $\textsc{Proc}(\mathcal{A}, \overline{\mathcal{A}}, \mathcal{F})$.

By a similar argument as in Case 2 in the proof of Lemma 5, replacing the two job sets $\mathcal{B}, \overline{\mathcal{B}}$ by the two job sets $\overline{\mathcal{A}}, \mathcal{A}$, we conclude that the makespan of the schedule $\pi^1$ is achieved on the machine $M_3$, $C_{\max}^{\pi^1} = P(\mathcal{F}) + Q(\mathcal{O}) + p_1 - Q(\overline{\mathcal{A}})$. Combining Eq. (1) with the assumption that $p_1 - \epsilon\mathrm{OPT}^1 < Q(\overline{\mathcal{A}})$, we have

$$C_{\max}^{\pi^1} < P(\mathcal{F}) + Q(\mathcal{O}) + \epsilon\mathrm{OPT}^1 \leq C_{\max}^* + \epsilon\mathrm{OPT}^1 < (1+\epsilon)C_{\max}^*,$$

where the last inequality follows from $\mathrm{OPT}^1 \leq Q(\mathcal{O}) \leq C_{\max}^*$. This finishes the proof of the lemma. □

**Theorem 1.** *The algorithm $A(\epsilon)$ is a Poly$(n, 1/\epsilon)$-time $(1+\epsilon)$-approximation for the problem $M3 \mid prpt \mid C_{\max}$ when $p_1 \geq q_{\ell+1}$.*

*Proof.* First of all, the procedure $\textsc{Proc}(\mathcal{X}, \mathcal{Y}, \mathcal{F})$ on a bipartition $\{\mathcal{X}, \mathcal{Y}\}$ of the job set $\mathcal{O}$ takes $O(n \log n)$ time. Recall that the job set $\mathcal{A}$ is computed by a $(1+\epsilon)$-approximation for the Min-Knapsack problem, which takes a polynomial time in both $n$ and $1/\epsilon$; the other job set $\mathcal{B}$ is computed by a $(1-\epsilon)$-approximation for the Max-Knapsack problem, which also takes a polynomial time in both $n$ and $1/\epsilon$. The total running time of the algorithm $A(\epsilon)$ is thus polynomial in both $n$ and $1/\epsilon$ too.

When $Q(\mathcal{O}) \leq p_1$, or the job set $\mathcal{O} \setminus \mathcal{A}$ computed in Step 2.1 of the algorithm $A_1(\epsilon)$ has total processing time not less than $p_1$, the schedule constructed in the algorithm $A(\epsilon)$ is optimal by Lemmas 1 and 2. When $Q(\mathcal{O} \setminus \mathcal{A}) < p_1$, the smaller makespan between the two schedules $\pi^1$ and $\pi^2$ constructed by the algorithm $A(\epsilon)$ is less than $(1+\epsilon)$ of the optimum by Lemmas 5 and 6. Therefore, the algorithm $A(\epsilon)$ has a worst-case performance ratio of $(1+\epsilon)$. This finishes the proof of the theorem. □

## 4 A 4/3-Approximation for the Case Where $p_1 < q_{\ell+1}$

In this section, we present a 4/3-approximation algorithm for the $M3 \mid prpt \mid C_{\max}$ problem when $p_1 < q_{\ell+1}$, and we show that this ratio of 4/3 is asymptotically tight.

**Theorem 2.** *When $p_1 < q_{\ell+1}$, the $M3 \mid prpt \mid C_{\max}$ problem admits an $O(n \log n)$-time 4/3-approximation algorithm.*

*Proof.* Consider first the case where there are at least two open-shop jobs. Construct a permutation schedule $\pi$ in which the job processing order for $M_1$ is $\langle J_{\ell+3}, \ldots, J_n, \mathcal{F}, J_{\ell+1}, J_{\ell+2} \rangle$, where the jobs of $\mathcal{F}$ are processed in the LPT order; the job processing order for $M_2$ is $\langle J_{\ell+2}, J_{\ell+3}, \ldots, J_n, \mathcal{F}, J_{\ell+1} \rangle$; the job processing order for $M_3$ is $\langle J_{\ell+1}, J_{\ell+2}, J_{\ell+3}, \ldots, J_n, \mathcal{F} \rangle$. See Fig. 5 for an illustration, where the start processing time for $J_{\ell+3}$ on $M_2$ is $q_{\ell+1}$, and the start processing time for $J_{\ell+3}$ on $M_3$ is $2q_{\ell+1}$. One can check that the schedule $\pi$ is feasible when $p_1 < q_{\ell+1}$, and it can be constructed in $O(n \log n)$ time.

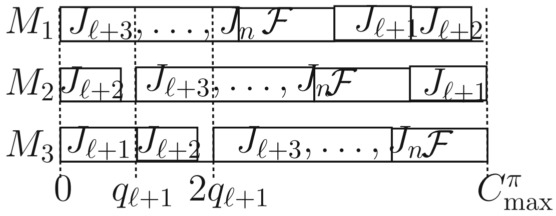


**Fig. 5.** A feasible schedule $\pi$ for the $M3 \mid prpt \mid C_{\max}$ problem with $p_1 < q_{\ell+1}$.

The makespan of the schedule $\pi$ is $C^{\pi}_{\max} = P(\mathcal{F}) + Q(\mathcal{O}) + q_{\ell+1} - q_{\ell+2}$. Combining this with Eq. (1), we have

$$C^{\pi}_{\max} \le P(\mathcal{F}) + Q(\mathcal{O}) + q_{\ell+1} \le \frac{4}{3} C^{*}_{\max}.$$

When there is only one open-shop job $J_{\ell+1}$, construct a permutation schedule $\pi$ in which the job processing order for $M_1$ is $\langle \mathcal{F}, J_{\ell+1} \rangle$, where the jobs of $\mathcal{F}$ are processed in the LPT order; the job processing order for $M_2$ is $\langle \mathcal{F}, J_{\ell+1} \rangle$; the job processing order for $M_3$ is $\langle J_{\ell+1}, \mathcal{F} \rangle$. If $P(\mathcal{F}) \le q_{\ell+1}$, then $\pi$ has makespan $3q_{\ell+1}$ and thus is optimal. If $P(\mathcal{F}) > q_{\ell+1}$, then $\pi$ has makespan $C^{\pi}_{\max} \le 2q_{\ell+1} + P(\mathcal{F}) \le \frac{4}{3} C^{*}_{\max}$. This finishes the proof of the theorem. □

*Remark 1.* Construct an instance in which $p_i = \frac{1}{\ell-1}$ for all $i = 1, 2, \ldots, \ell$, $q_{\ell+1} = 1$ and $q_i = \frac{1}{n-\ell-2}$ for all $i = \ell+2, \ell+3, \ldots, n$. Then for this instance, the schedule $\pi$ constructed in the proof of Theorem 2 has makespan $C^{\pi}_{\max} = 4 + \frac{1}{\ell-1}$; an optimal schedule has makespan $C^{*}_{\max} = 3 + \frac{1}{\ell-1} + \frac{1}{n-\ell-2}$ (see for an illustration in Fig. 6). This suggests that the approximation ratio of 4/3 is asymptotically tight for the algorithm in the proof of Theorem 2.

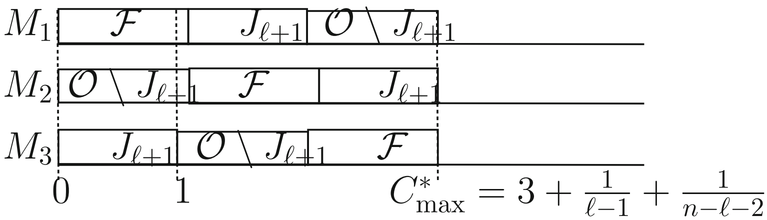


**Fig. 6.** An optimal schedule for the constructed instance of the $M3 \mid prpt \mid C_{\max}$ problem, in which $p_i = \frac{1}{\ell-1}$ for all $i = 1, 2, \ldots, n$, $q_{\ell+1} = 1$ and $q_i = \frac{1}{n-\ell-2}$ for all $i = \ell+2, \ell+3, \ldots, n$.

## 5 NP-Hardness for the Case Where $\mathcal{O} = \{J_n\}$ and $p_1 < q_n$

In this section, we show that the $M3 \mid prpt \mid C_{\max}$ problem with only one open-shop job is already NP-hard if this open-shop job is larger than any flow-shop job. We prove the NP-hardness through a reduction from the PARTITION problem [3], which is a well-known NP-complete problem.

**Theorem 3.** *The $M3 \mid prpt \mid C_{\max}$ problem with only one open-shop job is NP-hard if this open-shop job is larger than any flow-shop job.*

*Proof.* An instance of the PARTITION problem consists of a set $S = \{a_1, a_2, a_3, \ldots, a_m\}$ where each $a_i$ is a positive integer and $a_1+a_2+\ldots+a_m = 2B$, and the query is whether or not $S$ can be partitioned into two parts such that each part sums to exactly $B$.

Let $x > B$, and we assume that $a_1 \geq a_2 \geq \ldots \geq a_m$.

We construct an instance of the $M3 \mid prpt \mid C_{\max}$ problem as follows: there are in total $m+2$ flow-shop jobs, and their processing times are $p_1 = x, p_2 = x$, and $p_{i+2} = a_i$ for $i = 1, 2, \ldots, m$; there is only one open-shop job with processing time $q_{m+3} = B + 2x$. Note that the total number of jobs is $n = m + 3$, and one sees that the open-shop job is larger than any flow-shop job.

If the set $S$ can be partitioned into two parts $S_1$ and $S_2$ such that each part sums to exactly $B$, then we let $\mathcal{J}^1 = J_1 \cup \{J_i \mid a_i \in B_1\}$ and $\mathcal{J}^2 = J_2 \cup \{J_i \mid a_i \in B_2\}$. We construct a permutation schedule $\pi$ in which the job processing order for $M_1$ is $\langle \mathcal{J}^1, \mathcal{J}^2, J_{m+3} \rangle$, where the jobs of $\mathcal{J}^1$ and the jobs of $\mathcal{J}^2$ are processed in the LPT order, respectively; the job processing order for $M_2$ is $\langle \mathcal{J}^1, J_{m+3}, \mathcal{J}^2 \rangle$; the job processing order for $M_3$ is $\langle J_{m+3}, \mathcal{J}^1, \mathcal{J}^2 \rangle$. See Fig. 7 for an illustration, in which $J_1$ starts at time 0 on $M_1$, starts at time $x$ on $M_2$, and starts at time $B + 2x$ on $M_3$; $J_2$ starts at time $B + x$ on $M_1$, starts at time $2B + 4x$ on $M_2$, and starts at time $2B + 5x$ on $M_3$; $J_{m+3}$ starts at time 0 on $M_3$, starts at time $B + 2x$ on $M_2$, and starts at time $2B + 4x$ on $M_1$. The feasibility is trivial and its makespan is $C^{\pi}_{\max} = 3B + 6x$, suggesting the optimality.

Conversely, if the optimal makespan for the constructed instance is $3B + 6x = 3q_{m+3}$, then we will show next that $S$ admits a partition into two equal parts.

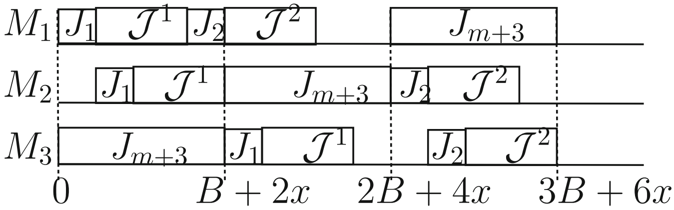


**Fig. 7.** A feasible schedule $\pi$ for the constructed instance of the $M3 \mid prpt \mid C_{\max}$ problem, when the set $S$ can be partitioned into two equal parts $S_1$ and $S_2$. The partition of the flow-shop jobs $\{\mathcal{J}^1, \mathcal{J}^2\}$ is correspondingly constructed. In the schedule, the jobs of $\mathcal{J}^1$ and the jobs of $\mathcal{J}^2$ are processed in the LPT order, respectively.

Firstly, we see that the second machine processing the open-shop job $J_{m+3}$ cannot be $M_1$, since otherwise $M_1$ has to process all the jobs of $\mathcal{F}$ before $J_{m+3}$, leading to a makespan greater than $3B + 6x$; the second machine processing the open-shop job $J_{m+3}$ cannot be $M_3$ either, since otherwise $M_3$ has no room to process any job of $\mathcal{F}$ before $J_{m+3}$, leading to a makespan larger than $3B + 6x$ too. Therefore, the second machine processing the open-shop job $J_{m+3}$ has to be $M_2$, see Fig. 8 for an illustration.

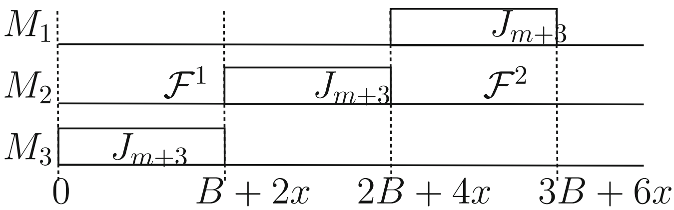


**Fig. 8.** An illustration of an optimal schedule for the constructed instance of the $M3 \mid prpt \mid C_{\max}$ problem with $\mathcal{O} = \{J_{m+3}\}$ and $q_{m+3} = B + 2x$. Its makespan is $3B + 6x = 3q_{m+3}$.

Denote the job subsets processed before and after the job $J_{m+3}$ on $M_2$ as $\mathcal{F}^1$ and $\mathcal{F}^2$, respectively. Since $x > B$, neither of $\mathcal{F}^1$ and $\mathcal{F}^2$ may contain both $J_1$ and $J_2$, which have processing times $x$. It follows that $\mathcal{F}^1$ and $\mathcal{F}^2$ each contains exactly one of $J_1$ and $J_2$, and subsequently $P(\mathcal{F}^1) = P(\mathcal{F}^2) = B + x$. Therefore, the jobs of $\mathcal{J}^1 \setminus \{J_1, J_2\}$ have a total processing time of exactly $B$, suggesting a subset of $S$ sums to exactly $B$. This finishes the proof of the theorem. $\square$

## 6 Concluding Remarks

In this paper, we studied the three-machine proportionate mixed shop problem $M3 \mid prpt \mid C_{\max}$. We presented first an FPTAS for the case where $p_1 \geq q_{\ell+1}$; and then proposed a 4/3-approximation algorithm for the other case where $p_1 < q_{\ell+1}$, for which we also showed that the performance ratio of 4/3 is asymptotically tight. The $F3 \mid prpt \mid C_{\max}$ problem is polynomial-time solvable; we showed an interesting hardness result that adding only one open-shop job to the job set makes the problem NP-hard if the open-shop job is larger than any flow-shop job.

We believe that when $p_1 < q_{\ell+1}$, the $M3 \mid prpt \mid C_{\max}$ problem can be better approximated than 4/3, and an FPTAS is perhaps possible. Nevertheless, a first step towards such an FPTAS is to design an FPTAS for the special case where there is only one open-shop job and the open-shop job is larger than any flow-shop job.

**Acknowledgements.** LL is supported by the CSC Grant 201706315073 and the Fundamental Research Funds for the Central Universities Grant No. 20720160035. GN is supported by the NSFC Grant 71501045, the NSF of Fujian Province Grant 2016J01332 and the Education Department of Fujian Province. YC and AZ are supported by the NSFC Grants 11771114 and 11571252; YC is also supported by the CSC Grant 201508330054. RG and GL are supported by the NSERC Canada; GL is also supported by the NSFC Grant 61672323.

## References

1. Brucker, P.: Scheduling Algorithms. Springer, Berlin (1995). https://doi.org/10.1007/978-3-662-03088-2
2. Chin, F.Y., Tsai, L.L.: On $j$-maximal and $j$-minimal flow shop schedules. J. ACM **28**, 462–476 (1981)
3. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, San Francisco (1979)
4. Graham, R.L., Lawler, E.L., Lenstra, J.K., Kan, R.: Optimization and approximation in deterministic sequencing and scheduling: a survey. Annu. Discret. Math. **5**, 287–326 (1979)
5. Kellerer, H., Pferschy, U.: Improved dynamic programming in connection with an FPTAS for the knapsack problem. J. Comb. Optim. **8**, 5–11 (2004)
6. Kellerer, H., Pferschy, U., Pisinger, D.: Knapsack Problems. Springer, Berlin (2004). https://doi.org/10.1007/978-3-540-24777-7
7. Koulamas, C., Kyparisis, G.J.: The three-machine proportionate open shop and mixed shop minimum makespan problems. Eur. J. Oper. Res. **243**, 70–74 (2015)
8. Liu, C.Y., Bulfin, R.L.: Scheduling ordered open shops. Comput. Oper. Res. **14**, 257–264 (1987)
9. Liu, L., et al.: Approximation algorithms for the three-machine proportionate mixed shop scheduling. CoRR 1809.05745 (2018)
10. Masuda, T., Ishii, H., Nishida, T.: The mixed shop scheduling problem. Discret. Appl. Math. **11**, 175–186 (1985)
11. Ow, P.S.: Focused scheduling in proportionate flowshops. Manag. Sci. **31**, 852–869 (1985)
12. Panwalkar, S., Smith, M.L., Koulamas, C.: Review of the ordered and proportionate flow shop scheduling research. Nav. Res. Logist. **60**, 46–55 (2013)
13. Shakhlevich, N., Sotskov, Y.N., Werner, F.: Shop-scheduling problems with fixed and non-fixed machine orders of the jobs. Ann. Oper. Res. **92**, 281–304 (1999)

14. Shakhlevich, N.V., Sotskov, Y.N.: Scheduling two jobs with fixed and nonfixed routes. Computing **52**, 17–30 (1994)
15. Shakhlevich, N.V., Sotskov, Y.N., Werner, F.: Complexity of mixed shop scheduling problems: a survey. Eur. J. Oper. Res. **120**, 343–351 (2000)
16. Strusevich, V.A.: Two-machine super-shop scheduling problem. J. Oper. Res. Soc. **42**, 479–492 (1991)