J Comb Optim (2013) 25:123–134
DOI 10.1007/s10878-011-9423-1

Two-stage proportionate flexible flow shop to minimize the makespan

Byung-Cheon Choi · Kangbok Lee

Published online: 30 November 2011

Abstract We consider a two-stage flexible flow shop problem with a single machine at one stage and m identical machines at the other stage, where the processing times of each job at both stages are identical. The objective is to minimize the makespan. We describe some optimality conditions and show that the problem is NP-hard when m is fixed. Finally, we present an approximation algorithm that has a worst-case performance ratio of $\frac{5}{4}$ for $m = 2$ and $\frac{\sqrt{1+m^2}+1+m}{2m}$ for $m \geq 3$.

Keywords Scheduling · Proportionate flexible flow shop · Computational complexity · Approximation algorithm

1 Introduction

We consider a two-stage proportionate flexible flow shop scheduling problem that can be stated as follows. There are n jobs to be processed through two stages in series such that one stage has one machine while the other has m identical parallel machines. Jobs can be processed on any machine at the stage with multiple machines. The processing times of each job at both stages are identical. The objective is to find a schedule for n jobs that minimizes the overall completion time (makespan), denoted by $C_{\max}$. Let our problem be referred to as *Problem P*.

B.-C. Choi
Department of Business Administration, Chungnam National University, 79 Daehangno, Yuseong-gu, Daejeon 305-704, Korea
e-mail: polytime@cnu.ac.kr

K. Lee (✉)
Department of Supply Chain Management & Marketing Sciences, Rutgers Business School, The State University of New Jersey, 1 Washington Park, Newark, NJ 07102, USA
e-mail: kangblee@business.rutgers.edu

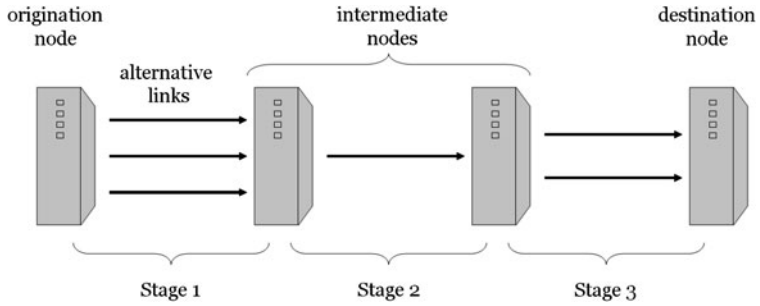

Fig. 1 A computer network with alternative links

The proportionate flexible flow shop problem has a practical application in data transmission in computer networks. We consider a computer network consisting of a series of nodes, including the origination, destination, and several intermediate nodes (see Fig. 1). The process of sending data from one node to the next is as follows. After completely receiving the data from the previous node, the current node checks its correctness and transmits it to the next node via one of alternative links. Most of the processing time in sending data is mainly spent transmitting data, and here the time to send data between two consecutive nodes is almost equal to the transmission time, which is a function of the number of bytes of data and the link's bandwidth. When the bandwidths of all the links over the overall network are the same, the processing times of each job at all stages are identical. Thus, this application can be modeled as a proportionate flexible flow shop problem such that

- The set of parallel links between two consecutive nodes corresponds to one stage;
- Each link corresponds to one machine at the corresponding stage;
- Each unit of data corresponds to each job;
- The data transmission time between two consecutive nodes corresponds to the processing time of the corresponding job.

We use the following notation throughout the paper. Let n denote the number of jobs, and m the number of machines at the stage with multiple machines. Let p_j refer to the processing time of job j. Let $\sigma = (\sigma_{k,i})$ denote a schedule where $\sigma_{k,i}$ is the sequence of jobs processed on machine i at stage k. Let job $\sigma_{k,i}(j)$ refer to the job that is in the jth position on machine i at stage k under schedule σ. Let $C_{\max}(\sigma)$ denote the makespan of schedule σ. Let $S_{i,j}(\sigma)$ and $C_{i,j}(\sigma)$ be the starting and completion times, respectively, of job j at stage i under schedule σ, $i = 1, 2$, $j = 1, \ldots, n$. If stage 1 has one machine, $\sigma = (\sigma_{1,1}; \sigma_{2,1}, \ldots, \sigma_{2,m})$; otherwise $\sigma = (\sigma_{1,1}, \ldots, \sigma_{1,m}; \sigma_{2,1})$. Using the classification scheme (Graham et al. 1979), we denote Problem P as $FF(1, m) \mid p_{ij} = p_j \mid C_{\max}$ for the first case and $FF(m, 1) \mid p_{ij} = p_j \mid C_{\max}$ for the second case, where $p_{i,j}$ denotes the processing time of job j at stage i. Throughout the paper, we assume that $p_1 \leq \cdots \leq p_n$.

The contributions of the paper are to provide a formal proof for the NP-completeness of Problem P with fixed m and present an approximation algorithm with the worst-case analysis. Note that the NP-completeness of the problem of minimizing the makespan on two identical parallel machines does not imply the NP-completeness

of Problem P, because in Problem P the processing times of each job in both stages are the same. For the non-proportionate case (i.e., when the processing times of a job at both stages are arbitrary), $(2-\frac{1}{m})$ is the best-known worst-case performance ratio (Lee and Vairaktarakis 1994). For Problem P, we present an approximation algorithm whose worst-case performance ratio is 5/4 for $m=2$ and $\frac{\sqrt{1+m^2}+1+m}{2m}$ for $m \geq 3$.

The rest of the paper is organized as follows. Section 2 discusses the related literature. Section 3 presents some optimality conditions for Problem P. Section 4 shows that Problem P is NP-hard. Section 5 provides an approximation algorithm and analyzes the worst-case performance ratio of the algorithm.

2 Literature review

In a proportionate flow shop environment, each stage has a single machine, and the processing times of job j over all stages are equal to p_j (Ow 1985; Pinedo 2008). Under this environment, the optimal makespan is $(\sum_{j=1}^{n} p_j + (m-1)\max\{p_1, \ldots, p_n\})$ and every sequence of the jobs is optimal (Pinedo 2008). Shakhlevich et al. (1998) showed that the problem of minimizing the total weighted completion time in a proportionate flow shop is solvable in time $O(n^2)$.

Linn and Zhang (1999), Ruiz et al. (2010) and Ribas et al. (2010) reviewed the state of the art in a flexible flow shop. To minimize the makespan for the two-stage case, Lee and Vairaktarakis (1994) developed an approximation algorithm with a worst-case performance ratio of $2-\frac{1}{\max\{m_1,m_2\}}$, where m_1 and m_2 are the numbers of machines at stages 1 and 2, respectively. Hoogeveen et al. (1996) showed that two-stage problems with and without preemption both are NP-hard in the strong sense. Schuurman and Woeginger (2000) designed a polynomial time approximation scheme (PTAS) for the two-stage problem such that the numbers of machines in both stages are part of the input. Thornton and Hunsucker (2004) considered a two-stage case with no intermediate storage and, from an experimental point of view, developed a better heuristic than those known.

Minimizing the makespan for a case with more than two stages has also been considered. Soewandi and Elmaghraby (2001) studied the problem with three stages and devised an approximation algorithm whose worst-case performance ratio is $\frac{10}{3}-\frac{1}{\max\{m_1,m_2\}}-\frac{1}{3m_3}$, where m_i is the number of machines at stage i, $i=1,2,3$. Koulamas and Kyparisis (2007) considered a c-stage case such that c is an odd number, and developed an approximation algorithm with a worst-case performance ratio of $c-\frac{1}{\max\{m_1,m_2\}}-\frac{1}{\max\{m_3,m_4\}}-\cdots-\frac{1}{\max\{m_{c-4},m_{c-3}\}}-\frac{1}{\max\{m_{c-2},m_{c-1},m_c\}}$, where m_i is the number of machines at stage i, $i=1,2,\ldots,c$. Jansen and Sviridenko (2000) developed a PTAS for the case with a fixed number of stages and a variable number of machines, and Sevastyanov (2008) improved the running time. Riane et al. (1998) considered the three-stage case with one machine in stages 1 and 3 and two machines in stage 2, and presented two heuristics that can be applied to realistic problems.

Little research has been conducted on the proportionate flexible flow shop so far. Pinedo (2008) considered the problem of minimizing the total completion time and showed that the shortest processing time (SPT) rule is optimal if each stage has at

least as many machines as the preceding stage. Huang and Shiau (2008) and Shiau et al. (2008) considered a problem in which the objective is to minimize the total weighted completion time and proposed a column generation approach and a hybrid constructive genetic algorithm, respectively.

3 Optimality conditions

This section presents some optimality conditions.

Theorem 1 *Let σ^* be an optimal schedule. Then*

(i) *$FF(1,m) \mid p_{ij} = p_j \mid C_{\max}$ has an optimal schedule such that if $S_{1,k}(\sigma^*) > S_{1,l}(\sigma^*)$, then $S_{2,k}(\sigma^*) \geq S_{2,l}(\sigma^*)$;*
(ii) *$FF(m,1) \mid p_{ij} = p_j \mid C_{\max}$ has an optimal schedule such that if $C_{1,k}(\sigma^*) \geq C_{1,l}(\sigma^*)$, then $C_{2,k}(\sigma^*) > C_{2,l}(\sigma^*)$.*

Proof (i) Suppose that there exists an optimal schedule such that job l is started immediately before job k at stage 1 and job k is started before job l at stage 2. Then, since the stage 2 operation of job k can be started after the stage 1 operation of job k is completed, the stage 2 operation of job l can be started after the stage 1 operation of job k is completed. Thus, we can construct a new feasible schedule by interchanging the positions of jobs k and l at stage 1 without increasing the makespan. By applying this procedure repeatedly, Theorem 1 can be attained.

(ii) Suppose that there exists an optimal schedule such that job l is completed before job k at stage 1 and job k is completed before job l at stage 2. Note that jobs k and l may be processed on different machines at stage 1.

Then, since the stage 2 operation of job k can be started after the stage 1 operation of job k is completed, the stage 2 operation of job l can be started after the stage 1 operation of job k is completed. Thus, we can construct a new feasible schedule by inserting job l immediately before k at stage 2 and shifting the jobs between jobs k and l to the right by p_l. Then, the new schedule is feasible and the makespan does not increase. By applying this procedure repeatedly, Theorem 1 can be attained. □

Theorem 2 *Let $\sigma^*_{1,1} = (\sigma^*_{k,i})$ be an optimal schedule. Then,*

(i) *$FF(m,1) \mid p_{i,j} = p_j \mid C_{\max}$ has an optimal schedule such that the jobs in $\sigma^*_{1,i}$ are processed in the SPT order, $i = 1, 2, \ldots, m$;*
(ii) *$FF(1,m) \mid p_{i,j} = p_j \mid C_{\max}$ has an optimal schedule such that the jobs in $\sigma^*_{2,i}$ are processed in the longest processing time (LPT) order, $i = 1, 2, \ldots, m$.*

Proof (i) Suppose that there exist jobs k and l such that $p_k < p_l$ and job l is processed before job k on the same machine at stage 1. By Theorem 1, job l is processed before job k at stage 2. We can construct a new schedule by inserting job k immediately before job l at both stages. Note that since $p_k < p_l$, the new schedule guarantees that the stage 1 operation of job k is completed before the stage 2 operation of job k is started. Also, the jobs between jobs k and l at stages 1 and 2 are shifted to the right

by p_k. Thus, the new schedule is feasible and the makespan does not increase. By applying this argument repeatedly, Theorem 2 can be obtained. The proof is complete.

(ii) By the reversibility property of flow shops, this immediately holds from (i). □

Henceforth, throughout the paper, we only consider schedules satisfying Theorems 1 and 2.

4 Computational complexity of Problem P

This section shows that Problem P is NP-hard. The decision version of Problem P is stated as follows: Given a threshold λ, is there a schedule σ such that $C_{\max}(\sigma) \leq \lambda$?

To prove the NP-completeness of Problem P, we introduce the partition problem, which is stated as follows: Given a set of n integers $\{a_1, a_2, \ldots, a_n\}$ such that $\sum_{j=1}^{n} a_j = A$, is there a subset $S \subset \{1, 2, \ldots, n\}$ such that $\sum_{j \in S} a_j = \frac{A}{2}$?

Theorem 3 *The decision version of Problem P is NP-complete when m is fixed.*

Proof Consider $FF(1, m) \mid p_{i,j} = p_j \mid C_{\max}$. It is clear that the decision version is in NP. Given an instance of the partition problem, we can construct an instance of $FF(1, m) \mid p_{i,j} = p_j \mid C_{\max}$ as follows. There is a set of $(n + 2m - 2)$ jobs, $\{1, 2, \ldots, n + 2m - 2\}$ such that

$$p_j = \begin{cases} a_j & \text{for } j = 1, \ldots, n, \\ 2^{n+m-j} B & \text{for } j = n+1, \ldots, n+m, \\ \frac{A}{2} & \text{for } j = n+m+1, \ldots, n+2m-2, \end{cases}$$

and $\lambda = 2^m B + \frac{A}{2}$, where let $B = \frac{A}{2}m$. This reduction can be done in polynomial time. Let the set of jobs be divided as follows:

- $J^1 := \{1, \ldots, n\}$;
- $J^2 := \{n+1, \ldots, n+m\}$;
- $J^3 := \{n+m+1, \ldots, n+2m-2\}$.

Suppose that there is a subset S such that $\sum_{j \in S} a_j = \frac{A}{2}$. Then, we can construct a schedule σ such that

- $\sigma_{1,1} = (n+1, n+2, \ldots, n+m, S, \bar{S}, n+m+1, \ldots, n+2m-2)$;
- $\sigma_{2,1} = (n+1, S)$;
- $\sigma_{2,2} = (n+2, \bar{S})$;
- $\sigma_{2,i} = (n+i, n+m+i-2)$, $i = 3, \ldots, m$;
- Jobs in S and $\bar{S}$ are processed in an arbitrary order,

where $\bar{S} = \{1, \ldots, n\} \setminus S$ (see Fig. 2). Thus, $C_{\max}(\sigma) = \lambda$.

Suppose that there exists a schedule σ such that $C_{\max}(\sigma) \leq \lambda$.

Claim 1 In schedule σ, job $(n+i)$ is processed before jobs in $\{n+i+1, n+i+2, \ldots, n+m\}$ at stage 1 for $i = 1, \ldots, m-1$.

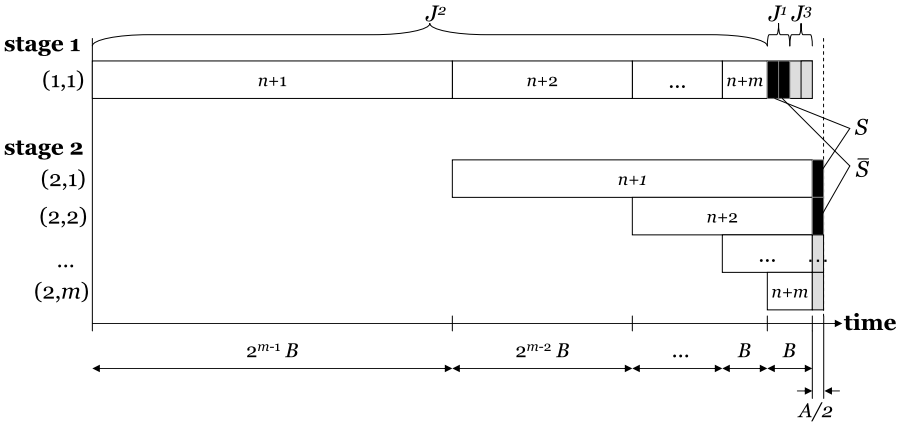

Fig. 2 The makespan of schedule σ

Proof Suppose Claim 1 holds for $i = 1, \ldots, k-1$ and does not hold when $i = k$. Then there exists a job $(n+l)$ processed before job $(n+k)$ at stage 1 such that $k < l \leq m$. Since $p_{n+l} \geq B$ and the starting time of job $(n+l)$ is larger than or equal to $\sum_{j=1}^{k-1} 2^{m-j} B$,

$$C_{\max}(\sigma) \geq C_{2,n+k}(\sigma) \geq \sum_{j=1}^{k-1} 2^{m-j} B + p_{n+l} + 2p_{n+k}$$
$$= (2^m - 2^{m-k+1})B + p_{n+l} + 2p_{n+k} \geq 2^m B + B > \lambda.$$

This is a contradiction. The proof is complete. □

It is observed from Claim 1 that

$$C_{2,j}(\sigma) \geq 2^m B \quad \text{for } j \in J^2. \tag{1}$$

Claim 2 In schedule σ, all jobs in J^2 are processed on different machines at stage 2.

Proof Suppose that there are jobs $(n+k)$ and $(n+l)$ that are processed on the same machine at stage 2, where $1 \leq k < l \leq m$. By inequality (1) and $p_{n+l} \geq B$,

$$C_{\max}(\sigma) \geq S_{2,n+l}(\sigma) + p_{n+l} \geq C_{2,n+k}(\sigma) + p_{n+l} \geq 2^m B + B > \lambda.$$

This is a contradiction. The proof is complete. □

By Claim 2, without loss of generality, we can assume that job $(n+i)$ is processed on machine i at stage 2 for each $i = 1, \ldots, m$.

Claim 3 In schedule σ, all jobs in $J^1 \cup J^3$ are processed after job $(n+m-1)$ at stage 1.

Proof Let J' be a set of jobs in $J^1 \cup J^3$ that are processed before job $(n+m-1)$ at stage 1. Let J'' be a set of jobs in $J^1 \cup J^3$ that are processed between jobs $(n+m-1)$ and $(n+m)$ at stage 1. Suppose that $J' \neq \emptyset$. Then, it is observed that

- $C_{2,n+j}(\sigma) \geq 2^m B$, $j = 1, \ldots, m-2$;
- $C_{2,n+m-1}(\sigma) \geq 2^m B + \sum_{j \in J'} p_j$;
- $C_{2,n+m}(\sigma) \geq 2^m B + \sum_{j \in J' \cup J''} p_j$.

By the above inequalities, we have

$$\sum_{j=1}^{m} \big(\lambda - C_{2,n+j}(\sigma)\big) \leq \frac{mA}{2} - 2\sum_{j \in J'} p_j - \sum_{j \in J''} p_j. \tag{2}$$

Moreover, since $C_{\max}(\sigma) \leq \lambda$, the total processing times of jobs in $(J^1 \cup J^3) \setminus (J' \cup J'')$ should be less than or equal to $\sum_{j=1}^{m}(\lambda - C_{2,n+j}(\sigma))$; that is,

$$\sum_{j \in (J^1 \cup J^3) \setminus (J' \cup J'')} p_j \leq \sum_{j=1}^{m} \big(\lambda - C_{2,n+j}(\sigma)\big). \tag{3}$$

Since $\sum_{j \in (J^1 \cup J^3) \setminus (J' \cup J'')} p_j = \frac{mA}{2} - \sum_{j \in J' \cup J''} p_j$ and $J' \neq \emptyset$, inequalities (2) and (3) cannot be satisfied simultaneously. Since this is a contradiction, $J' = \emptyset$. The proof is complete. □

Claim 4 In schedule σ, the total processing time of jobs in $J^1 \cup J^3$ that are scheduled on the same machine at stage 2 is at most $\frac{A}{2}$.

Proof It is observed from Claims 1 and 3 that $C_{2,n+j}(\sigma) = 2^m B$ for $j = 1, \ldots, m-1$. Thus, the total processing time of jobs in $J^1 \cup J^3$ that are scheduled on machine i at stage 2 should be at most $\frac{A}{2}$ for $i = 1, \ldots, m-1$.

Henceforth, we focus on jobs that are scheduled on machine m at stage 2. Let J' (or J'') be the set of jobs in $J^1 \cup J^3$ that are scheduled before (or after) job $(n+m)$ at stage 1 and scheduled on machine m at stage 2. Let job l be the last job in J''. Then,

$$C_{2,l}(\sigma) = C_{1,n+m-1}(\sigma) + \sum_{j \in J'} p_j + 2p_{n+m} + \sum_{j \in J''} p_j = 2^m B + \sum_{j \in J' \cup J''} p_j.$$

Since $C_{\max}(\sigma) \leq \lambda$, $\sum_{j \in J' \cup J''} p_j \leq \frac{A}{2}$. The proof is complete. □

By Claim 4 and $\sum_{j \in J^1 \cup J^3} p_j = \frac{A}{2}m$, the total processing time of jobs in $J^1 \cup J^3$ that are scheduled on machine i at stage 2 is exactly $\frac{A}{2}$ for $i = 1, \ldots, m$. This implies that there exists a subset $S \subset \{1, 2, \ldots, n\}$ such that $\sum_{j \in S} a_j = \frac{A}{2}$. The proof is complete. □

5 Approximation algorithm

This section only considers $FF(1,m)|p_{i,j} = p_j|C_{\max}$, because the results for $FF(1,m)|p_{i,j} = p_j|C_{\max}$ hold for $FF(m,1)|p_{i,j} = p_j|C_{\max}$ by the reversibility property of flow shops. Recall that $p_1 \leq \cdots \leq p_n$. Consider the following algorithm.

Algorithm LPT

Step 1 Sequence jobs by order of decreasing p_j at stage 1.
Step 2 Set $i = 1$ and $j = n$.

Step 2-1 Assign job j to machine i at stage 2. Set $i = i + 1$ and $j = j - 1$.
Step 2-2 If $i = m + 1$, then $i = 1$.
Step 2-3 If $j = 0$, then go to Step 3; otherwise go to Step 2-1.

Step 3 Shift the jobs to the left as much as possible at stage 2 while the completion time of a job at stage 1 is less than or equal to the starting time of the job at stage 2.

The time complexity of Algorithm LPT is $O(n \log n)$.

Numerical example Consider a 7-job two-stage flexible flow shop with one machine at stage 1 and two machines at stage 2 as follows:

j	1	2	3	4	5	6	7
p_j	1	2	3	4	8	9	10

Step 1: Set $\sigma^{LPT} = (7, 6, 5, 4, 3, 2, 1)$.
Step 2: Sequence jobs belonging to $\{1, 3, 5, 7\}$ and $\{2, 4, 6\}$ in LPT order on machines 1 and 2 at stage 2, respectively, that is, $\sigma_{2,1}^{LPT} = (7, 5, 3, 1)$ and $\sigma_{2,2}^{LPT} = (6, 4, 2)$.
Step 3: Jobs in $\{5, 6, 7\}$ at stage 1 and jobs in $\{1, 3, 5\}$ at stage 2 contribute the makespan. The makespan is

$$\begin{aligned} C_{\max}(\sigma^{LPT}) &= p_{1,7} + p_{1,6} + p_{1,5} + p_{2,5} + p_{2,3} + p_{2,1} \\ &= p_7 + p_6 + p_5 + p_5 + p_3 + p_1 = 39. \end{aligned}$$

Let $\sigma^* = (\sigma_{1,1}^*; \sigma_{2,1}^*, \ldots, \sigma_{2,m}^*)$ be the optimal schedule, and $\sigma^{LPT} = (\sigma_{1,1}^{LPT}; \sigma_{2,1}^{LPT}, \ldots, \sigma_{2,m}^{LPT})$ be the schedule obtained by Algorithm LPT. Note that by Step 3 of Algorithm LPT, there must exist job l such that $C_{1,l}(\sigma^{LPT}) = S_{2,l}(\sigma^{LPT})$ that is critical for the makespan. (In the above numerical example, job 5 is such a critical job.) Without loss of generality, assume that job l is processed on machine α at stage 2. Then,

$$C_{\max}(\sigma^{LPT}) = \sum_{j=l}^{n} p_j + p_l + \sum_{j \in S_\alpha} p_j, \tag{4}$$

where S_α is the set of jobs that are processed after job l on machine α at stage 2. Let S_i be the set of jobs that are assigned to machine i at stage 2 by Algorithm LPT after job l has been assigned to machine α for $i \in \{1, \ldots, m\} \setminus \{\alpha\}$. Note that $\bigcup_{i=1}^{m} S_i = \{1, 2, \ldots, l-1\}$. Then, it is observed that

$$\sum_{j \in S_i} p_j \geq \sum_{j \in S_\alpha} p_j \quad \text{for } i = 1, \ldots, m. \tag{5}$$

Inequality (5) implies

$$\sum_{j\in S_\alpha} p_j \le \frac{1}{m}\sum_{j=1}^{l-1} p_j. \tag{6}$$

Theorem 4 *For* $FF(1,m) \mid p_{ij} = p_j \mid C_{\max}$,

$$\frac{C_{\max}(\sigma^{LPT})}{C_{\max}(\sigma^*)} \le \frac{\sqrt{1+m^2}+1+m}{2m}.$$

Proof By equation (4) and inequality (6),

$$C_{\max}(\sigma^{LPT}) \le \sum_{j=l}^{n} p_j + p_l + \frac{1}{m}\sum_{j=1}^{l-1} p_j. \tag{7}$$

Claim $\sum_{j=l}^{n} p_j + p_l \le C_{\max}(\sigma^*)$.

Proof Let $\sigma^*_{1,1}(k)$ be the first job such that

$$\{l, l+1, \ldots, n\} \subset \{\sigma^*_{1,1}(1), \sigma^*_{1,1}(2), \ldots, \sigma^*_{1,1}(k)\}.$$

Note that job $\sigma^*_{1,1}(k)$ belongs to $\{l, l+1, \ldots, n\}$. Since $\sum_{j=1}^{k} p_{\sigma^*_{1,1}(j)} \ge \sum_{j=l}^{n} p_j$ and $p_{\sigma^*_{1,1}(k)} \ge p_l$,

$$C_{\max}(\sigma^*) \ge \sum_{j=1}^{k} p_{\sigma^*_{1,1}(j)} + p_{\sigma^*_{1,1}(k)} \ge \sum_{j=l}^{n} p_j + p_l.$$

The proof is complete. □

Let $\alpha = \sqrt{1+m^2}+1-m$. We consider two cases on the ratio of p_l and $\sum_{j=1}^{l-1} p_j$.

Case 1. $\alpha p_l \ge \sum_{j=1}^{l-1} p_j$.

By the claim above, inequality (7), and $2p_l \le C_{\max}(\sigma^*)$,

$$C_{\max}(\sigma^{LPT}) - C_{\max}(\sigma^*) \le \frac{1}{m}\sum_{j=1}^{l-1} p_j \le \frac{\alpha}{m} p_l \le \frac{\alpha}{2m} C_{\max}(\sigma^*).$$

Case 2. $\alpha p_l < \sum_{j=1}^{l-1} p_j$.

By $\sum_{j=1}^{n} p_j \le C_{\max}(\sigma^*)$ and inequality (7),

$$\begin{aligned} C_{\max}(\sigma^{LPT}) - C_{\max}(\sigma^*) &\le p_l + \frac{1}{m}\sum_{j=1}^{l-1} p_j - \sum_{j=1}^{l-1} p_j \le \left(\frac{1}{\alpha} + \frac{1}{m} - 1\right)\sum_{j=1}^{l-1} p_j \\ &\le \left(\frac{1}{\alpha} + \frac{1}{m} - 1\right) C_{\max}(\sigma^*). \end{aligned}$$

By Cases 1 and 2,

$$C_{\max}(\sigma^{LPT}) \le \left(1+\frac{\alpha}{2m}\right) C_{\max}(\sigma^*) \quad \text{and} \quad C_{\max}(\sigma^{LPT}) \le \left(\frac{1}{\alpha}+\frac{1}{m}\right) C_{\max}(\sigma^*).$$

Since $\alpha = \sqrt{1+m^2}+1-m$, we have

$$1+\frac{\alpha}{2m} = \frac{1}{\alpha}+\frac{1}{m} = \frac{\sqrt{1+m^2}+1+m}{2m}.$$

Thus,

$$\frac{C_{\max}(\sigma^{LPT})}{C_{\max}(\sigma^*)} \le \frac{\sqrt{1+m^2}+1+m}{2m}.$$

The proof is complete. □

Note that

$$1+\frac{1}{2m} < \frac{\sqrt{1+m^2}+1+m}{2m} < 1+\frac{1}{2m}+\frac{1}{4m^2}.$$

Consider an instance as follows: There is a set of $(mx+1)$ jobs consisting of a job with processing time $(mx+1)$ and mx jobs with processing time 1 for integer x. The optimal makespan $C_{\max}(\sigma^*)$ is $2(mx+1)$, while the makespan of the schedule by Algorithm LPT $C_{\max}(\sigma^{LPT})$ is $(2(mx+1)+x)$. When x goes to infinity, the performance ratio approaches $(1+\frac{1}{2m})$.

The performance ratio of Algorithm LPT for the above instance is $(1+\frac{1}{2m})$, implying that the worst-case performance ratio in Theorem 4 is almost tight if m is sufficiently large.

In the case for $m=2$, Algorithm LPT has a tight worst-case performance ratio of $1+\frac{1}{4}$.

Theorem 5 *For* $FF(1,2) \mid p_{ij} = p_j \mid C_{\max}$,

$$\frac{C_{\max}(\sigma^{LPT})}{C_{\max}(\sigma^*)} \le \frac{5}{4}.$$

Proof By equation (4) and inequality (6),

$$C_{\max}(\sigma^{LPT}) \le \sum_{j=l}^{n} p_j + p_l + \frac{1}{2}\sum_{j=1}^{l-1} p_j.$$

Since $p_l \le p_n$,

$$C_{\max}(\sigma^{LPT}) \le \left(\sum_{j=l}^{n} p_j + \frac{1}{2}\sum_{j=1}^{l} p_j\right) + \frac{1}{2}p_n. \tag{8}$$

Claim $(\sum_{j=l}^{n} p_j + \frac{1}{2}\sum_{j=1}^{l} p_j) \le C_{\max}(\sigma^*)$.

Proof Consider two cases.

Case 1: $p_l \le \sum_{j=1}^{l-1} p_j$

In this case,

$$\sum_{j=1}^{n} p_j - \left(\sum_{j=l}^{n} p_j + \frac{1}{2}\sum_{j=1}^{l} p_j\right) = \frac{1}{2}\left(\sum_{j=1}^{l-1} p_j - p_l\right) \ge 0.$$

Since $C_{\max}(\sigma^*) \ge \sum_{j=1}^{n} p_j$, $C_{\max}(\sigma^*) \ge (\sum_{j=l}^{n} p_j + \frac{1}{2}\sum_{j=1}^{l} p_j)$.

Case 2: $p_l > \sum_{j=1}^{l-1} p_j$

Since $p_l > \sum_{j=1}^{l-1} p_j$,

$$\sum_{j=l}^{n} p_j + \frac{1}{2}\sum_{j=1}^{l} p_j < \sum_{j=l}^{n} p_j + \frac{1}{2}p_l + \frac{1}{2}p_l = \sum_{j=l}^{n} p_j + p_l.$$

Recall that by the claim in the proof of Theorem 4, $\sum_{j=l}^{n} p_j + p_l \le C_{\max}(\sigma^*)$. Thus,

$$\sum_{j=l}^{n} p_j + \frac{1}{2}\sum_{j=1}^{l} p_j < C_{\max}(\sigma^*).$$

By Cases 1 and 2, the claim above holds. □

By the claim above and $p_n \le \frac{1}{2}C_{\max}(\sigma^*)$, (8) becomes

$$C_{\max}(\sigma^{LPT}) \le C_{\max}(\sigma^*) + \frac{1}{4}C_{\max}(\sigma^*) = \frac{5}{4}C_{\max}(\sigma^*).$$

This completes the proof. □

Note: By applying the argument in proof of Theorem 5 to the case with $m \ge 3$, we have

$$C_{\max}(\sigma^{LPT}) \le \left(\sum_{j=l}^{n} p_j + \frac{1}{m}\sum_{j=1}^{l} p_j\right) + \frac{m-1}{m}p_n \quad \text{and}$$

$$\sum_{j=l}^{n} p_j + \frac{1}{m}\sum_{j=1}^{l} p_j \le C_{\max}(\sigma^*).$$

By the above inequalities and $2p_n \le C_{\max}(\sigma^*)$, we have

$$\frac{C_{\max}(\sigma^{LPT})}{C_{\max}(\sigma^*)} \le \frac{3m-1}{2m}.$$

Since $\sqrt{1+m^2} < 1+m$ for $m \ge 3$, however,

$$\frac{3m-1}{2m} - \frac{\sqrt{1+m^2}+1+m}{2m} = \frac{2m-2-\sqrt{1+m^2}}{2m} > \frac{2m-2-(1+m)}{2m}$$
$$= \frac{m-3}{2m} \ge 0.$$

Thus, it is observed that for the case with $m \geq 3$, the bound obtained by the argument in proof of Theorem 5 is worse than the bound of Theorem 4.

References

Graham RL, Lawler EL, Lenstra JK, Rinnooy Kan AHG (1979) Optimization and approximation in deterministic sequencing and scheduling: a survey. Ann Discrete Math 5:287–326

Hoogeveen JA, Lenstra JK, Veltman B (1996) Preemptive scheduling in a two-stage multiprocessor flow shop is NP-hard. Eur J Oper Res 89:172–175

Huang YM, Shiau DF (2008) Combined column generation and constructive heuristic for a proportionate flexible flow shop scheduling. Int J Adv Manuf Technol 38:691–704

Jansen K, Sviridenko MI (2000) Polynomial time approximation schemes for the multiprocessor open and flow shop scheduling problem. Lect Notes Comput Sci 1770:455–565

Koulamas C, Kyparisis GJ (2007) A note on performance guarantees for sequencing three-stage flexible flowshops with identical machines to minimize makespan. IIE Trans 39:559–563

Lee CY, Vairaktarakis GL (1994) Minimizing makespan in hybrid flowshop. Oper Res Lett 16:149–158

Linn R, Zhang W (1999) Hybrid flow shop scheduling: a survey. Comput Ind Eng 37:57–61

Ow PS (1985) Focused scheduling in proportionate flowshops. Manag Sci 31:852–869

Pinedo M (2008) Scheduling: theory, algorithms and systems, 3rd edn. Springer, Berlin

Riane F, Artiba A, Elmaghraby SE (1998) A hybrid three-stage flowshop problem: efficient heuristics to minimize makespan. Eur J Oper Res 109:321–329

Ribas I, Leisten R, Framinan JM (2010) Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. Comput Oper Res 37(8):1439–1454

Ruiz R, Antonio J, Rodriguez V (2010) The hybrid flow shop scheduling problem. Eur J Oper Res 205:1–18

Schuurman P, Woeginger GJ (2000) A polynomial time approximation scheme for the two-stage multiprocessor flow shop problem. Theor Comput Sci 237:105–122

Sevastyanov SV (2008) An improved approximation scheme for the Johnson problem with parallel machines. J Appl Ind Math 2(3):406–420

Shakhlevich N, Hoogeveen H, Pinedo M (1998) Minimizing total weighted completion time in a proportionate flow shop. J Sched 1:157–168

Shiau DF, Cheng SC, Huang YM (2008) Proportionate flexible flow shop scheduling via a hybrid constructive genetic algorithm. Expert Syst Appl 34:1133–1143

Soewandi H, Elmaghraby SE (2001) Sequencing three-stage flexible flowshops with identical machines to minimize makespan. IIE Trans 33(11):985–994

Thornton HW, Hunsucker JL (2004) A new heuristic for minimal makespan in flow shops with multiple processors and no intermediate storage. Eur J Oper Res 152:96–114