An improved heuristic for two-machine flowshop scheduling with an availability constraint

T.C. Edwin Cheng\textsuperscript{a, *}, Guoqing Wang\textsuperscript{b,c}

\textsuperscript{a}Office of the Vice-President (Research and Postgraduate Studies) The Hong Kong Polytechnic University, Kowloon, Hong Kong
\textsuperscript{b}Department of Business Administration, Jinan University, Guangzhou, People’s Republic of China
\textsuperscript{c}Department of Management, The Hong Kong Polytechnic University, Kowloon, Hong Kong

Received 1 May 1998; received in revised form 1 March 2000

Abstract

In this paper we study the two-machine flowshop scheduling problem with an availability constraint on the first machine. We first show that the worst-case error bound $\frac{1}{2}$ of the heuristic provided by Lee [4] is tight. We then develop an improved heuristic with a worst-case error bound of $\frac{1}{3}$. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Flowshop scheduling; Heuristics; Error bound

1. Introduction

In this paper, we address the two-machine flowshop scheduling problem with an availability constraint on the first machine. We are given two machines $M_1$ and $M_2$, and a job set $S = \{J_1, \ldots, J_n\}$. Each job has to be processed first on $M_1$ and then on $M_2$. In our problem, $M_2$ is always available, but $M_1$ is unavailable for job processing during a certain period of time and the time interval of unavailability is assumed to be known in advance. All jobs are assumed resumable, i.e. if a job cannot finish before the unavailable period of a machine, then its processing may be interrupted and resumed at a later time when the machine becomes available again. The objective is to minimize the makespan.

The two-machine flowshop scheduling problem with an availability constraint is first studied by Lee [4]. He shows that the problem is NP-hard and develops a pseudo-polynomial dynamic programming algorithm to solve the problem optimally. When the availability constraint is imposed on $M_1$, he provides a heuristic H2 to tackle the problem and proves that its worst-case error bound is $\frac{1}{2}$. But the problem of whether the bound is tight is left open. He also provides another heuristic with a worst-case error bound of $\frac{1}{3}$ for a variant of the problem, where the availability constraint is imposed on $M_2$. For other works on flowshop scheduling with availability constraints, we refer the interested reader to Kubiat et al. [3], Lee [5], and Cheng and Wang [1].
In this paper, we first show that the bound $\frac{1}{2}$ is tight for H2. We then provide another heuristic to solve the problem and show that the heuristic has a worst-case error bound of $\frac{1}{3}$.

2. Notation

For the problem under consideration, we use the following notation.

- $S = \{J_1, \ldots, J_n\}$: the set of jobs,
- $M_1, M_2$: machine 1 and machine 2,
- $s_1, t_1$: $M_1$ is unavailable from $s_1$ to $t_1$, where $0 \leq s_1 \leq t_1$,
- $p_i, q_i$: processing times for $J_i$ on $M_1$ and $M_2$, respectively,
- $\sigma = (J_{a(1)}, \ldots, J_{a(n)})$: a permutation schedule, where $J_{a(i)}$ is the $i$th job in $\sigma$,
- $C_{\text{max}}(\sigma)$: the makespan of $\sigma$,
- $\sigma^*$: an optimal schedule,
- $C^*$: the optimal makespan.

Following the notation of Lee [4], we denote the problem under study as $F2|\text{r-a}(M_1)|C_{\text{max}}$, i.e. the makespan minimization problem in the two-machine flowshop with a resumable availability constraint on $M_1$.

It is well-known that the classical two-machine flowshop makespan minimization problem, $F2||C_{\text{max}}$, can be solved by Johnson’s rule [2]. Johnson’s rule can be implemented as follows:

**Johnson’s algorithm.** Divide $S$ into two disjoint subsets $A$ and $B$, where $A = \{J_i | q_i \geq p_i\}$ and $B = \{J_i | q_i < p_i\}$. Order the jobs in $A$ in nondecreasing order of $p_i$ and those in $B$ in nonincreasing order of $q_i$. Sequence jobs in $A$ first, followed by those in $B$.

Let $C^*$ be the optimal makespan for $F2||C_{\text{max}}$. It is evident that

$$C^* \leq C^*.$$  \hspace{1cm} (1)

3. An improved heuristic and its worst-case error bound

The heuristic H2 proposed by Lee for $F2/|\text{r-a}(M_1)|C_{\text{max}}$ is as follows:

**Heuristic H2**

(i) Use Johnson’s algorithm to schedule the jobs and let the corresponding schedule be $\sigma_1$.

(ii) Sequence jobs in nonincreasing order of $q_i/p_i$ and let the corresponding schedule be $\sigma_2$. Let $t$ be the starting time of the last busy period on $M_2$, namely $M_2$ is idle immediately before $t$. Let $J_k$ be the corresponding job that starts at $t$ on $M_2$.

(iii) Sequence all jobs in the same order as that in Step (ii) except that we make $J_k$ the first job in the sequence. Let the corresponding schedule be $\sigma_3$.

(iv) Choose the schedule with the minimum makespan from the above three schedules. Let $C_{H2} = \min(C_{\text{max}}(\sigma_1), C_{\text{max}}(\sigma_2), C_{\text{max}}(\sigma_3))$.

It is shown in [4] that $(C_{H2} - C^*)/C^* \leq \frac{1}{2}$. Below, we complete the analysis of the algorithm by demonstrating that this bound is tight. Consider a problem instance with $n = 3$. Let $p_1 = 1$, $q_1 = a + 1$, $p_2 = a$, $q_2 = a^2$, $p_3 = a + 2$, and $q_3 = (a + 2)^2$, $s_1 = a$, $t_1 = a^2 + a$, and $s_2 = t_2 = 0$, where $a > 1$.

It is clear that Step (i) will result in sequence $\sigma_1 = \langle J_1, J_2, J_3 \rangle$ with $C_{\text{max}}(\sigma_1) = 3a^2 + 5a + 5$ (see Fig. 1(a)). Steps (ii) and (iii) will result in $\sigma_2 = \sigma_3 = \langle J_3, J_1, J_2 \rangle$, with $C_{\text{max}}(\sigma_2) = C_{\text{max}}(\sigma_3) = 3a^2 + 6a + 7$ (see Fig. 1(b)). Hence, $C_{H2} = 3a^2 + 5a + 5$. However, the optimal solution is $\sigma^* = \langle J_2, J_1, J_3 \rangle$ with $C^* = 2a^2 + 6a + 7$ (see Fig. 1(c)). We see that $(C_{H2} - C^*)/C^*$ goes to $\frac{1}{2}$ as $a$ approaches infinity.

In what follows, we present an improved heuristic that finds a schedule for $F2/|\text{r-a}(M_1)|C_{\text{max}}$ that is at most 4/3 times the optimal value.

**Heuristic H1**

(i) Let $J_a$ and $J_b$ be two jobs with the largest and the second largest processing time on $M_2$, respectively, i.e. $\min\{q_i, q_b\} = q_i$ for $i = 1, \ldots, n$, $i \neq a$, and $i \neq b$.

(ii) Use Johnson’s algorithm to schedule the jobs and let the corresponding schedule be $\sigma_1$.

(iii) Sequence jobs in nonincreasing order of $q_i/p_i$. Let the corresponding schedule be $\sigma_2$. Let $t$ be the starting time of the latest busy period on $M_2$ in $\sigma_2$, namely $M_2$ is idle immediately before $t$. Let $J_k$ be the corresponding job that starts at $t$ on $M_2$.

(iv) Sequence the jobs in the same sequence as that in Step (iii) except that $J_a$ and $J_b$ are scheduled as the first two jobs such that the completion time of the last one is minimized. Let the corresponding schedule be $\sigma_3$.
If \( t > s_1 \) and \( p_k \leq s_1 \), then construct \( \sigma_4 \) by maintaining all jobs in the same order they appear in \( \sigma_2 \), with the exception of \( J_k \) which is moved to position \( u \) in which it finishes by time \( s_1 \) and the job in position \( u + 1 \) finishes after \( s_1 \) on \( M_1 \).

Let \( S_1 = \{ J_i \mid \frac{q_i}{p_i} \geq 1 \text{ and } p_i < p_k \} \) and \( S_2 = S \setminus (S_1 \cup \{ J_k \}) \). Let \( \sigma_5 = (S_1, J_k, S_2) \), where all jobs in \( S_1 \) are scheduled in nonincreasing order of \( q_i/p_i \) and jobs in \( S_2 \) are scheduled according to Johnson’s rule.

Choose the schedule with the minimum makespan from the above five schedules.

**Lemma 2.** Let \( T_{\sigma_2(w)} \) be the completion time of any job \( J_{\sigma_2(w)} \) on \( M_1 \) in \( \sigma_2 \). Given an arbitrary feasible schedule \( \sigma \), let \( J_{\sigma(v)} \) be the last job which finishes no later than time \( T_{\sigma_2(w)} \) on \( M_1 \). We have

(i) \( \sum_{j=1}^{w} q_{\sigma(j)} \leq \sum_{j=1}^{n} q_{\sigma_2(j)} \),
(ii) \( \sum_{j=1}^{w} q_{\sigma(j)} \geq \sum_{j=1}^{n} q_{\sigma_2(j)} \).

**Proof** It is clear that \( \sum_{j=1}^{w} p_{\sigma(j)} \leq \sum_{j=1}^{n} p_{\sigma_2(j)} \) and \( \sum_{j=t+1}^{n} p_{\sigma(j)} \geq \sum_{j=1}^{n} p_{\sigma_2(j)} \). Since all jobs are sequenced in nonincreasing order of \( q_i/p_i \) in \( \sigma_2 \), we can easily check the results.

**Lemma 3.** For schedule \( \sigma_2 \) and job \( J_k \) defined in Step (iii) of Heuristic HI, the following inequalities hold:

(i) \( C_{\text{max}}(\sigma_2) \leq C^* + q_k \),
(ii) If \( t \leq s_1 \) or \( t \geq t_1 + p_k \), then \( C_{\text{max}}(\sigma_2) \leq C^* + p_k \).

**Proof** Assume that \( J_{\sigma_2(w)} = J_k \), then \( C_{\text{max}}(\sigma_2) = t + \sum_{j=w}^{n} q_{\sigma_2(j)} \).

(i) Let \( J_{\sigma_2(v)} \) be the last job which finishes no later than time \( t \) on \( M_1 \) in \( \sigma^* \). From Lemma 2(ii), we have \( \sum_{j=t+1}^{w} q_{\sigma(j)} \geq \sum_{j=1}^{w} q_{\sigma_2(j)} \). Hence, we have \( C_{\text{max}}(\sigma_2) \leq C^* + q_k \).

(ii) Let \( L \) be the total idle time on \( M_2 \) in \( \sigma_2 \). It is clear that \( L = t - \sum_{j=1}^{w-1} q_{\sigma_2(j)} \). Let \( J_{\sigma_2(v')} \) be the last job...
which finishes no later than time $T_{s_2(w-1)}$ on $M_1$, and $L^*$ the total idle time on $M_2$ in $\sigma^*$. From Lemma 2(i), we know that $\sum_{j=1}^{w} q_{\sigma^*(j)} = \sum_{j=1}^{w} q_{s_2(j)}$. When $t \leq s_1$ or $t - p_k \geq t_1$, it is clear that $L^* \geq t - p_k - \sum_{j=1}^{w} q_{s_2(j)}$. Hence, we have $L^* \geq L - p_k$, and so $C_{\text{max}}(\sigma_2) = C^* + p_k$.

**Theorem 1.** $(C_{\text{III}} - C^*)/C^* \leq \frac{1}{3}$.

**Proof.** It suffices to consider the case with $\sum p_i > s_1$. Note that $C_{\text{max}}(\sigma_1) \leq C^* + t_1 - s_1$. Hence, if $t_1 - s_1 \leq C^*/3$ we are done. Therefore, in the remainder of the proof, we are only interested in the case with $t_1 - s_1 > C^*/3$. Let $\tilde{S} = \{J_i | q_i > C^*/3, i = 1, \ldots, n\}$. It is obvious that $|\tilde{S}| \leq 2$. When $|\tilde{S}| = 0$, from Lemma 3(i), we have $C_{\text{max}}(\sigma_2) = C^* + q_k < 4C^*/3$. Hence, we only need to consider the following two cases.

**Case 1:** $|\tilde{S}| = 2$

In this case, we have $\tilde{S} = \{J_a, J_b\}$. Let us consider Step (iv). Let $C^*(\tilde{S})$ denote the optimal completion time of the jobs in $\tilde{S}$ in $\sigma_3$. It is clear that $C^*(\tilde{S}) \leq C^*$. Let $t'$ be the starting time of the last busy period on $M_2$ in $\sigma_3$, and $J_l$ be the corresponding job that starts at $t'$ on $M_2$. If $J_l = J_a$ or $J_b$, we have

$$C_{\text{max}}(\sigma_3) = C^*(\tilde{S}) + \sum_{J_i \in \tilde{S}} q_i - q_a - q_b$$

$$\leq C^* + C^*/3 \leq 4C^*/3;$$

otherwise, we have

$$C_{\text{max}}(\sigma_3) \leq t' + \sum_{J_i \in \tilde{S}} q_i - q_a - q_b$$

$$\leq \sum_{J_i \in \tilde{S}} p_i + (t_1 - s_1) + C^*/3$$

$$\leq 4C^*/3.$$

**Case 2:** $|\tilde{S}| = 1$

Let us consider $\sigma_2$. Since $C_{\text{max}}(\sigma_2) \leq C^* + q_k$ from Lemma 2(i), it suffices to consider $q_k = q_a > C^*/3$, for otherwise we are done. Now, we need to consider the following two subcases.

1. $t \leq s_1$

From Lemma 3(ii), we have $C_{\text{max}}(\sigma_2) \leq C^* + p_k$. If $p_k \leq C^*/3$, then we are done. Now suppose that $p_k > C^*/3$. Note that we must have $q_k < 2C^*/3$ and

$$\sum_{J_i \in S \setminus \{J_k\}} p_i \leq C^* - p_k - (t_1 - s_1) \leq C^*/3. \tag{2}$$

It is clear that schedule $\sigma_2$ will result in the same makespan for both $F2/r-\sigma(M_1)/C_{\text{max}}$ and $F2//C_{\text{max}}$ in this subcase, and the optimal makespan $C^*$ for $F2/r-\sigma(M_1)/C_{\text{max}}$ is no less than the optimal makespan $C^*$ for $F2//C_{\text{max}}$. Let $A' = \{J_i | q_i \geq p_i, i = 1, \ldots, n, i \neq k\}$ and $B' = S \setminus (A' \cup \{J_k\})$. It is not difficult to see that there exists an optimal solution for $F2//C_{\text{max}}$ in which all jobs in $A'$ are scheduled before $J_k$ and all jobs in $B'$ are scheduled after $J_k$ according to Johnson’s rule. Thus, we have

$$C^* \geq C_{\text{max}}(\sigma_2) \geq \sum_{J_i \in A'} p_i + p_k + q_k \sum_{J_i \in B'} q_i \tag{3}$$

and so

$$\sum_{J_i \in B'} q_i \leq C^* - p_k - q_k < C^*/3. \tag{4}$$

We first suppose that $q_k \geq p_k$. Let $N$ be the set of jobs which follow $J_k$ in $\sigma_2$ and satisfy $q_i \geq p_i$. It is clear that

$$\sum_{J_i \in N} q_i \leq p_k \sum_{J_i \in N} p_i \tag{5}$$

Since all jobs are scheduled in nonincreasing order of $q_i/p_i$ in $\sigma_2$, it is clear that $\sigma_2 = \langle A' \setminus N, J_k, N, B' \rangle$.

From (2)–(5), we have

$$C_{\text{max}}(\sigma_2) = \sum_{J_i \in A' \setminus N} p_i + p_k + q_k + \sum_{J_i \in N} q_i + \sum_{J_i \in B'} q_i$$

$$\leq \sum_{J_i \in A' \setminus N} p_i + p_k + q_k$$

$$+ \frac{q_k}{p_k} \sum_{J_i \in N} p_i + \sum_{J_i \in B'} q_i$$

$$\leq \sum_{J_i \in A'} p_i + p_k + q_k + \sum_{J_i \in B'} q_i$$

$$+ \left(\frac{q_k}{p_k} - 1\right) \sum_{J_i \in N} p_i$$

$$< C^* + \sum_{J_i \in N} p_i$$

$$< 4C^*/3.$$
Now suppose that \( q_k < p_k \). Let \( N \) be the set of jobs which follow \( J_k \) in \( \sigma_2 \). From (2)–(4), we have

\[
C_{\max}(\sigma_2) = \sum_{J \in S \setminus \{A\}} p_i + p_k + q_k + \sum_{J \in N} q_i \\
\leq \sum_{J \in S \setminus \{A\}} p_i + p_k + q_k + \sum_{J \in N} q_i \\
\leq \sum_{J \in S \setminus \{A\}} p_i + C^* \\
< 4C^*/3.
\]

2. \( t > s_1 \). We consider the following two situations.

(a) There exists an optimal schedule \( \sigma^* \) such that \( J_k \) finishes before \( s_1 \) on \( M_1 \).

Let us focus on schedule \( \sigma_4 \) obtained in Step (v). Let \( t' \) be the starting time of the last busy period on \( M_2 \), and \( J_{\alpha(t)} \) be the corresponding job that starts at \( t' \) on \( M_2 \). Whenever \( v < u \) or \( v > u \), following the same argument as that for Lemma 3, we can easily show that \( C_{\max}(\sigma_4) - C^* \leq q_{\alpha(v)} \leq C^*/3 \). Now we focus on \( v = u \), i.e. \( J_{\alpha(v)} = J_k \).

Let \( L \) be the total idle time on \( M_2 \) in \( \sigma_4 \). We have

\[
L = p_k - \sum_{j=1}^{u-1} (q_{\sigma(j)} - p_{\sigma(j)}).
\]

If \( p_k \leq C^*/3 \), then all jobs preceding \( J_k \) in \( \sigma_4 \) satisfy the conditions

\[
q_i \geq q_k \geq p_k > 1.
\]

Hence, we have

\[
L < p_k \leq C^*/3.
\]

If \( p_k > C^*/3 \), note again that we have

\[
\sum_{J \in S \setminus \{A\}} p_i \leq C^* - (t_1 - s_1) - p_k \leq C^*/3.
\]

Let \( J_{\sigma^*(\alpha')} = J_k \) in \( \sigma^* \). Then we have

\[
L^* \geq p_k - \sum_{j=1}^{u'-1} (q_{\sigma^*(j)} - p_{\sigma^*(j)}),
\]

and so

\[
L \leq L^* + \left( \sum_{j=1}^{u-1} p_{\sigma(j)} - \sum_{j=1}^{u'-1} p_{\sigma^*(j)} \right) + \left( \sum_{j=1}^{u'-1} q_{\sigma^*(j)} - \sum_{j=1}^{u-1} q_{\sigma(j)} \right).
\]

If \( \sum_{j=1}^{u-1} p_{\sigma(j)} \geq \sum_{j=1}^{u'-1} p_{\sigma^*(j)} \), it is clear that \( \sum_{j=1}^{u-1} q_{\sigma(j)} \geq \sum_{j=1}^{u'-1} q_{\sigma^*(j)} \) as all jobs except \( J_k \) are sequenced in nondecreasing order of \( q_i/p_i \) in \( \sigma_4 \). From (7), we have

\[
L \leq L^* + \sum_{j=1}^{u-1} p_{\sigma(j)} < L^* + C^*/3.
\]

If \( \sum_{j=1}^{u-1} p_{\sigma(j)} < \sum_{j=1}^{u'-1} p_{\sigma^*(j)} \), we have \( \sum_{j=1}^{u-1} q_{\sigma(j)} \geq \sum_{j=1}^{u'-1} q_{\sigma^*(j)} \) and so

\[
L < L^* + \sum_{j=1}^{u'-1} q_{\sigma^*(j)} - \sum_{j=1}^{u-1} q_{\sigma(j)} \\
< L^* + q_{\sigma(u+1)} < L^* + C^*/3.
\]

(b) There exists no optimal schedule such that \( J_k \) finishes before \( s_1 \) on \( M_1 \).

It is clear that \( p_k < C^*/3 \) and so \( p_k < q_k \); otherwise, we have \( C^* \geq t_1 - s_1 + p_k + q_k > C^* \), a contradiction. Let us focus on schedule \( \sigma_5 \) obtained in Step (vi). We first show that there exists an optimal solution in which all jobs in \( S_1 \) are scheduled before \( J_k \), and all jobs in \( S_2 \) are scheduled after \( J_k \) in this situation. Let \( \sigma^* \) be an optimal schedule, and assume, without loss of generality, that \( \sigma^* \) satisfies Lemma 1. Then it is immediately clear that any job \( J_i \in S_1 \) which finishes after \( t_1 \) on \( M_1 \) is scheduled before \( J_k \) in \( \sigma^* \). This also means that all jobs in \( S_2 \) are scheduled before \( J_k \) in \( \sigma^* \). As it is also clear that any job \( J_i \) such that \( p_i \geq p_k \) which finishes after \( t_1 \) can be scheduled after \( J_k \), now suppose that there is a job \( J_i \) such that \( p_i \geq p_k \) finishes before \( s_1 \) in \( \sigma^* \). Since \( q_i < q_k \), it is easy to see that interchanging \( J_i \) and \( J_k \) in \( \sigma^* \) will not increase the makespan. This means that there exists an optimal solution in which \( J_k \) finishes before \( s_1 \) on \( M_1 \), a contradiction. Again, as it is clear that any job \( J_i \) such that \( p_i > q_i \) and \( p_i < p_k \) which finishes after \( s_1 \) on \( M_1 \) follows \( J_k \) in \( \sigma^* \), we suppose that there are
some jobs such that $p_i > q_i$ and $p_i < p_k$ which finish before $s_1$ on $M_1$ in $\sigma^*$. Let $J_f$ be the last job which finishes before $s_1$ on $M_1$ in $\sigma^*$. It is clear that $p_i > q_j$, as $\sigma^*$ satisfies Lemma 1. Since $p_i < p_k < q_k$, it is not difficult to see that moving $J_f$ to the position immediately after $J_k$ will not increase the makespan. Repeating the same process, we see that there is an optimal schedule in which all jobs in $S_2$ are scheduled after $J_k$.

As $J_k$ finishes after $t_1$ on $M_1$ in $\sigma^*$, and so all jobs in $S_2$ must also finish after $t_1$ on $M_1$, and it is optimal to assign the jobs by Johnson’s rule. Let $C^*(S_1)$ be the completion time of the last job in $S_1$ in the optimal solution. Then we have

$$C^* = C^*(S_1) + L + q_k + \sum_{J_i \in S_2} q_i,$$

where $L$ is the total idle time on $M_2$ between the last job of $S_1$ and the last job of $S_2$ in the optimal solution.

Let $C(S_1) = C^*(S_1) + \Delta$ be the completion time of the last job in $S_1$ in $\sigma_5$. From Lemma 3(i), we know that

$$\Delta \leq \max\{q_i | J_i \in S_1\} < C^*/3.$$

Fig. 2. Solution of Step (ii); (b) solution of Steps (iii) and (iv); (c) solution of Step (v); (d) solution of Step (vi) and (e) optimal solution.
From (8) and (9), we have
\[ C_{\max}(\sigma_5) = C(S_1) + \max\{0, L - A\} + q_k + \sum_{J_i \in S_2} q_i \]
\[ < C^*(S_1) + L + q_k + \sum_{J_i \in S_2} q_i + C^*/3 \]
\[ < 4C^*/3 \]
The proof is complete. \(\square\)

**Remark**
(i) The time complexity of the algorithm is \(O(n \log n)\).

(ii) Although we are not able to show that the bound is tight, the following instance with \(n = 3\) shows that the bound cannot be better than \(1/3\). Consider a problem with \(p_1 = a - 1\), \(q_1 = (a - 1)^2\), \(p_2 = a\), \(q_2 = a^2\), \(p_3 = a + 1\), and \(q_3 = 2(a + 1)^2\), \(s_1 = 2a\), and \(t_1 = 2a^2 + 2a\), where \(a \gg 1\). It is clear that Step (ii) will result in schedule \(\sigma_1 = \langle J_1, J_2, J_3 \rangle\) with
\[ C_{\max}(\sigma_1) = 5a^2 + 7a + 2 \] (see Fig. 2(a)). Both Steps (iii) and (iv) will result in schedule \(\sigma_2 = \sigma_3 = \langle J_3, J_2, J_1 \rangle\) with
\[ C_{\max}(\sigma_2) = C_{\max}(\sigma_3) = 5a^2 + 2 \] (see Fig. 2(b)).
Step (v) will result in schedule \(\sigma_4 = \langle J_2, J_3, J_1 \rangle\) with
\[ C_{\max}(\sigma_4) = 6a^2 + 4a + 4 \] (see Fig. 2(c)).
Step (vi) will result in schedule \(\sigma_5 = \langle J_2, J_1, J_3 \rangle\) with
\[ C_{\max}(\sigma_5) = 5a^2 + 7a + 2 \] (see Fig. 2(d)).
Hence, \(C_{II} = 5a^2 + 2\). However, the optimal solution is \(\sigma^* = \langle J_3, J_1, J_2 \rangle\) with \(C^* = 4a^2 + 3a + 4\) (see Fig. 2(d)). Hence, \((C_{II} - C^*)/C^*\) goes to \(1/4\) as \(a\) approaches infinity.

**Acknowledgements**

This research was supported in part by The Hong Kong Polytechnic University under grant number G-YW16. We are grateful to an anonymous referee for his constructive comments on earlier versions of the paper.

**References**