J Math Model Algor (2014) 13:511–522
DOI 10.1007/s10852-013-9245-5

A PTAS for a Particular Case of the Two-machine Flow Shop with Limited Machine Availability

Hatem Hadda

Received: 17 May 2013 / Accepted: 6 November 2013 / Published online: 16 November 2013

Abstract In this paper we develop a polynomial-time approximation scheme for a particular case of the two-machine flow shop scheduling problem with several availability constraints on the second machine under the resumable scenario.

Keywords Flow shop · Scheduling · Availability constraint · Approximation scheme

Mathematics Subject Classification (2010) 90B35 scheduling theory · Deterministic · 68W40 analysis of algorithms

1 Introduction

This paper tackles a special case of the two-machine flow shop scheduling problem with several availability constraints (holes for short) on the second machine. The objective is to find a schedule of n given jobs that minimizes the maximum completion time (i.e., the makespan). Each job J_i is composed by two operations (O_{iA} and O_{iB}), which have to be processed on two machines A and B. Each machine can process at most one job at a time. Machine B is assumed to be unavailable during q holes, and the precise time of each hole is known in advance. Three scenarios are possible when an operation is interrupted by a hole. In the *semiresumable* (sr) model the operation will have to partially restart when the machine becomes available again. In the *resumable* (r) model the operation can be continued without any penalty, and in the *nonresumable* model (nr) the operation needs to totally restart. In this paper all jobs are supposed to be resumable. We consider the case where the starting time of the last hole is such that $s_q < C^{\star}_{\max}$ where $C^{\star}_{\max}$ is the optimal makespan. The problem is strongly NP-hard and will be denoted $F2|h(0,q), r, s_q < C^{\star}_{\max}|C_{\max}$.

When a scheduling problem is classified as NP-hard, research focuses on developing approximation algorithms with some guarantees on the quality of the obtained results. In

H. Hadda (✉)
LISI, INSAT-Tunis, Centre Urbain Nord B.P. N676, 1080 Tunis Cedex, Tunisia
e-mail: hatem.hadda@esti.rnu.tn

this respect, a ρ-approximation algorithm is a polynomial-time algorithm that constructs a solution with a makespan that is at most ρ times the optimal one. In the same context, a polynomial-time approximation scheme (PTAS) is a family of $(1+\varepsilon)$-approximation algorithms for $\varepsilon > 0$. If in addition the running time is polynomial in $1/\varepsilon$, the algorithm is said fully polynomial-time approximation scheme (FPTAS). Note that if a problem is strongly NP-hard then it does not admit a FPTAS (unless P=NP) [3].

The $F2||C_{\max}$ is the only non-trivial variant of the flow shop problem that is solvable in polynomial time [10]. Indeed the $F3||C_{max}$ and the two-stage hybrid flow shop problem are both strongly NP-hard [2, 9]. Furthermore, it is shown that unless P=NP, there does not exist a ρ-approximation algorithm for the flow shop problem with $\rho < 5/4$ [17]. In [16] a PTAS is proposed for the two stage hybrid flow shop problem. In the view of the approximability hardness of the general flow shop problem, research focused on the two-stage configurations. Therefore, the works that addressed the availability constraints have mainly considered the two-machine flow shop problem.

The two-machine flow shop with a single hole was first considered by Lee [13, 14] who proposed a $\left(\frac{3}{2}\right)$-approximation algorithm for $F2|h(0,1), sr|C_{\max}$, and a 2-approximation algorithm for $F2|h(1,0), sr|C_{\max}$. For the nonresumable scenario, a $\left(\frac{3}{2}\right)$-approximation algorithm has been proposed for $F2|h(1,0), nr|C_{\max}$ [7]. Furthermore, it has been shown that the two-machine flow shop with a single hole admits a PTAS under the semiresumable scenario [11] and a FPTAS under the resumable scenario [15].

Considering a variable number of holes, the $F2|h(q,0), r|C_{\max}$ is the only configuration that may admit a fixed factor approximation. For this problem, a $\left(\frac{4}{3}\right)$-approximation algorithm and a polynomially solvable case were proposed in [5]. It has also been shown that it admits a PTAS [6]. Finally, a $\left(\frac{4}{3}\right)$-approximation algorithm has been proposed for a particular case of $F2|h(0,q), r|C_{\max}$ [4]. In the same context, three basic approximation algorithms are proposed in [1] for the two-stage hybrid flow shop with several holes, and in [8] several approximation algorithms are developed for the two-stage assembly flow shop problem under an availability constraint.

The remainder of this paper is organized as follows. Section 2 introduces some notations. Section 3 introduces a PTAS for the $F2|h(0,q), r, s_q < C^\star_{\max}|C_{\max}$ problem. Finally, Section 4 provides some concluding remarks.

2 Notation

We will use the following notation.

$J = \{J_1, \ldots, J_n\}$: Set of jobs.
a_i, b_i: Processing times for $J_i \in J$ on A and B respectively.
$\pi = \langle J_{\pi(1)}, \ldots, J_{\pi(n)} \rangle$: Job permutation, where $J_{\pi(i)}$ is the ith job in π.
q: Number of holes.
$s_k,\ t_k$: Start and finish time of hole k, $1 \le k \le q$. We assume that $s_1 < s_2 < \cdots < s_q$.
$h_k = t_k - s_k$: Length of hole k, $1 \le k \le q$.
$S_{ij}(\pi)$ and $C_{ij}(\pi)$: Start and finish time of operation O_{ij}, $i \in \{1, \ldots, n\}$, $j \in \{A, B\}$ in schedule π.
$C_{\max}(\pi)$: Makespan of π.
$\pi^\star$: An optimal schedule.
$C^\star_{\max}$: Optimal makespan.

We also define $a(Q) = \sum_{J_k \in Q} a_k$, $b(Q) = \sum_{J_k \in Q} b_k$ for a non-empty set Q of jobs.

For a given job J_z in π, we define $H_z(\pi) = \sum_I h_k$, where $I = \{h_k \mid s_k > S_{zB}\}$. The reference to schedule π will be dropped whenever no confusion can arise. Furthermore, all operations are assumed to start as early as possible.

We now recall the following rules for the two-machine flow shop problem.

Johnson's rule [10]: J_i precedes J_j if $\min(a_i, b_j) \leq \min(a_j, b_i)$.
Ratio rule (RR): J_i precedes J_j if $b_i/a_i > b_j/a_j$. If $b_i/a_i = b_j/a_j$ break tie arbitrarily.

As explained before, it is assumed that

$$s_q < C^\star_{\max}. \tag{1}$$

This will guarantee that the last hole will affect all schedules including the optimal ones. As $C^\star_{\max}$ is unknown, it is possible to use instead a lower bound LB. We give here two possible values for LB. The first one is given by $LB = b(J) + \sum_{i=1}^{q-1} h_i$. For the second one, schedule the n jobs according to Johnson's rule and consider the corresponding makespan (without considering the holes).

Note that it is sufficient to consider permutation schedules [12]. In order to determine the makespan of a given schedule π, we have to search for the job $J_z = J_{\pi(u)}$ which starts the last busy period on machine B. One of the following two conditions must be realized (see Fig. 1):

(Condition 1) $S_{zB} = C_{zA}$. Hence

$$\begin{aligned} C_{\max}(\pi) &= C_{zA} + \sum_{i=u}^{n} b_{\pi(i)} + H_z(\pi) \\ &= \sum_{i=1}^{u} a_{\pi(i)} + \sum_{i=u}^{n} b_{\pi(i)} + H_z(\pi). \end{aligned} \tag{2}$$

(Condition 2) There exists a hole h_r such that $s_r \leq C_{zA} < t_r$. Hence

$$C_{\max}(\pi) = t_r + \sum_{i=u}^{n} b_{\pi(i)} + H_z(\pi). \tag{3}$$

3 Approximation Scheme

This section introduces a PTAS for $F2|h(0,q), r, s_q < C^\star_{\max}|C_{\max}$. Recall that the best approximation algorithm known for this problem guarantees a relative worst-case error

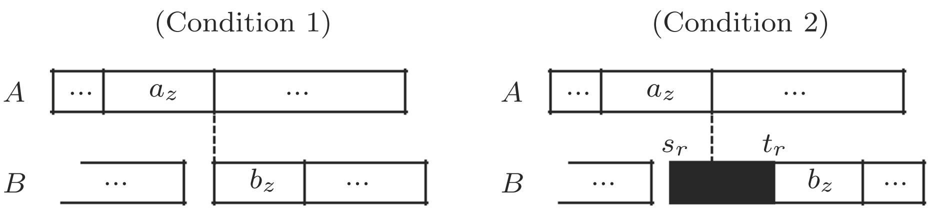

Fig. 1 Possible configurations for the makespan

bound of 4/3 [4]. The proposed PTAS is inspired from the one introduced in [6] for the $F2|h(q, 0), r|C_{\max}$ problem.

Given any fixed $\varepsilon > 0$, we will designate by a *big job*, a job J_i such that $a_i \geq \varepsilon C^{\star}_{\max}$. The rest of the jobs will be referred as *small jobs*. Furthermore, the set of all big jobs will be denoted $\overline{J} = \{J_i | a_i \geq \varepsilon C^{\star}_{\max}\}$ and $r = |\overline{J}|$.

The idea of the algorithm consists on testing all possible placements for the big jobs. More specifically, for each r-permutation $\sigma = \langle\sigma(1), \sigma(2), \ldots, \sigma(r)\rangle$ of r elements from the set $\{1, 2, \ldots, n\}$, the jobs of $\overline{J}$ will be scheduled in positions $\sigma(i)$, $1 \leq i \leq r$. The rest of the jobs will be scheduled in the $(n - r)$ left positions according to *RR*. Considering all possible $\frac{n!}{(n-r)!}$ r-permutations, we will obtain the desired solution.

Assume that $\varepsilon < 1$ and that the number of jobs is sufficiently large, e.g., $n > \lceil\frac{1}{\varepsilon}\rceil$. The PTAS is described in Algorithm 1.

Algorithm 1: *Algorithm* H^{ε}

(i) Sequence the jobs according to RR. Call the resulting schedule π_0. And let $C_H = C_{\max}(\pi_0)$.
Let $k = 1$.
for $p = 1$ **to** $\lceil\frac{1}{\varepsilon}\rceil$ **do**
  Let S be the set containing the p jobs with the largest processing times on machine A, i.e., $S = \{J_i | a_i \geq a_j \quad \forall J_j \in J\backslash S\}$ and $|S| = p$.
  foreach *p-permutation* $\sigma = \langle\sigma(1), \ldots, \sigma(p)\rangle$ *of p elements from the set* $\{1, \ldots, n\}$ **do**
    (ii) Schedule the jobs of S in positions $\sigma(i)$, $1 \leq i \leq p$.
    (iii) Sequence the rest of the jobs $(J\backslash S)$ in the vacant positions according to RR. Call the resulting schedule π_k.
    Let $C_H = \min\{C_H, C_{\max}(\pi_k)\}$ and $k = k + 1$.
  endforeach
endfor

As an illustration, consider an example with $n = 5$ and $\varepsilon = 1/2$ where the jobs are indexed according to *RR* (i.e. $\pi_0 = \langle J_1, J_2, J_3, J_4, J_5\rangle$). Suppose that J_1 and J_3 are such that $a_1 \geq a_3 \geq a_i$ for $i \in \{2, 4, 5\}$.

For $p = 1$ we get $S = \{J_1\}$. Steps (ii) and (iii) generate the following schedules: $\langle \boldsymbol{J_1}, J_2, J_3, J_4, J_5\rangle$, $\langle J_2, \boldsymbol{J_1}, J_3, J_4, J_5\rangle$, $\langle J_2, J_3, \boldsymbol{J_1}, J_4, J_5\rangle$, $\langle J_2, J_3, J_4, \boldsymbol{J_1}, J_5\rangle$, and $\langle J_2, J_3, J_4, J_5, \boldsymbol{J_1}\rangle$.

For $p = 2$ we get $S = \{J_1, J_3\}$. There are 20 2-permutations from the set $\{1, 2, 3, 4, 5\}$. Consequently, steps (ii) and (iii) generate the following schedules:

$$\langle \boldsymbol{J_1}, \boldsymbol{J_3}, J_2, J_4, J_5\rangle, \langle \boldsymbol{J_1}, J_2, \boldsymbol{J_3}, J_4, J_5\rangle, \langle \boldsymbol{J_1}, J_2, J_4, \boldsymbol{J_3}, J_5\rangle, \langle \boldsymbol{J_1}, J_2, J_4, J_5, \boldsymbol{J_3}\rangle,$$
$$\langle \boldsymbol{J_3}, \boldsymbol{J_1}, J_2, J_4, J_5\rangle, \langle \boldsymbol{J_3}, J_2, \boldsymbol{J_1}, J_4, J_5\rangle, \langle \boldsymbol{J_3}, J_2, J_4, \boldsymbol{J_1}, J_5\rangle, \langle \boldsymbol{J_3}, J_2, J_4, J_5, \boldsymbol{J_1}\rangle,$$
$$\langle J_2, \boldsymbol{J_1}, \boldsymbol{J_3}, J_4, J_5\rangle, \langle J_2, \boldsymbol{J_1}, J_4, \boldsymbol{J_3}, J_5\rangle, \langle J_2, \boldsymbol{J_1}, J_4, J_5, \boldsymbol{J_3}\rangle,$$
$$\langle J_2, \boldsymbol{J_3}, \boldsymbol{J_1}, J_4, J_5\rangle, \langle J_2, \boldsymbol{J_3}, J_4, \boldsymbol{J_1}, J_5\rangle, \langle J_2, \boldsymbol{J_3}, J_4, J_5, \boldsymbol{J_1}\rangle,$$
$$\langle J_2, J_4, \boldsymbol{J_1}, \boldsymbol{J_3}, J_5\rangle, \langle J_2, J_4, \boldsymbol{J_1}, J_5, \boldsymbol{J_3}\rangle,$$
$$\langle J_2, J_4, \boldsymbol{J_3}, \boldsymbol{J_1}, J_5\rangle, \langle J_2, J_4, \boldsymbol{J_3}, J_5, \boldsymbol{J_1}\rangle,$$
$$\langle J_2, J_4, J_5, \boldsymbol{J_1}, \boldsymbol{J_3}\rangle \text{and} \langle J_2, J_4, J_5, \boldsymbol{J_3}, \boldsymbol{J_1}\rangle.$$

Steps (ii) and (iii) can be executed in $O(n)$ and have to be repeated $\frac{n!}{(n-p)!} < n^p$ times for $1 \leq p \leq \lceil \frac{1}{\varepsilon} \rceil$. This means that the number of operations is bounded by $O(n) \sum_{p=1}^{\lceil \frac{1}{\varepsilon} \rceil} n^p$. Therefore, the time complexity of Algorithm H^ε is $O(n^{\lceil \frac{1}{\varepsilon} \rceil + 1})$.

Given a schedule π_k generated by H^ε, we designate by a *block* of jobs, a sequence of big jobs directly succeeding each other. A big job is said to be *introductory* if it is the first in a block of jobs.

Before giving the worst-case error bound of H^ε, we establish the following two lemmas which will be used in the subsequent analysis.

Lemma 1 *Given an optimal solution $\pi^\star$, and supposing that $\overline{J} \neq \emptyset$, there exists a schedule π generated by H^ε such that:*

(i) The big jobs appear in the same order as that in $\pi^\star$.
(ii) Every big job J_x verifies $S_{xA}(\pi^\star) \leq S_{xA}(\pi) < C_{xA}(\pi^\star)$.
(iii) Every introductory job $J_y = J_{\pi(v)}$ of π realizes one of the following two conditions:

- *J_y is the first job in $\pi^\star$.*
- *$S_{\pi(v-1)A}(\pi) < S_{yA}(\pi^\star) \leq S_{yA}(\pi)$.*

Proof Since $a(J)$ is a lower bound on $C^\star_{\max}$, and by definition of $\overline{J}$, we have that $r = |\overline{J}| \leq \lceil \frac{1}{\varepsilon} \rceil$. As Algorithm H^ε tests all possible values of $1 \leq p \leq \lceil \frac{1}{\varepsilon} \rceil$, then necessarily it will get a configuration where $S = \overline{J}$. That configuration is considered in the reminder of the present proof (i.e., $p = r = |\overline{J}|$).

Considering that Algorithm H^ε tests all p-permutations for the positions of jobs in S, it follows that there will be $\binom{n}{p} = \frac{n!}{p!(n-p)!}$ solutions in which the order of the big jobs will be the same as in $\pi^\star$ which satisfies (i). We now establish that one of these solutions realizes conditions (ii) and (iii).

Consider solution $\pi = \langle \underline{J \backslash S}, \underline{S} \rangle$ where $\underline{S}$ is composed by the jobs of S scheduled in the same order as in $\pi^\star$, and $\underline{J \backslash S}$ is composed by the jobs of $J \backslash S$ sequenced according to *RR*. This schedule is clearly one of the solutions that are generated by H^ε. In the following, we are going to rearrange the jobs of $\underline{S}$ starting from the beginning of the subsequence.

Let J_y be the first job in $\underline{S}$. If J_y is scheduled in the first position in $\pi^\star$ then we consider the solution π' obtained from π by placing J_y in the first position and by shifting forwards all the jobs initially scheduled before J_y in π. Otherwise, let $J_{\pi(v)}$ be the first job in π such that $S_{\pi(v)A}(\pi) \geq S_{yA}(\pi^\star)$ (see Fig. 2). Consider the solution π' obtained from π by scheduling J_y in position v and by shifting forwards the jobs initially scheduled between $J_{\pi(v-1)}$ and J_y in π. In this case schedule π' is such that $S_{\pi'(v-1)A}(\pi') = S_{\pi(v-1)A}(\pi) < S_{yA}(\pi^\star) \leq S_{\pi(v)A}(\pi) = S_{yA}(\pi')$. In either cases the obtained solution π' is generated by H^ε, and J_y verifies (iii) (note that J_y is an introductory job). Let $\pi = \pi'$.

Consider now the first job J_x of the subsequence $\underline{S} \backslash \{J_y\}$ and let $J_{\pi(v)}$ be the first job in π such that $S_{\pi(v)A}(\pi) \geq S_{xA}(\pi^\star)$. Two cases arise:

If $J_{\pi(v-1)}$ is a big job, then consider solution π' obtained from π by scheduling J_x in the end of the block of jobs containing $J_{\pi(v-1)}$. In this case, J_x is not an introductory job and verifies the first inequality of condition (ii) as $S_{xA}(\pi') \geq S_{\pi(v)A}(\pi) \geq S_{xA}(\pi^\star)$.

Otherwise $J_{\pi(v-1)}$ is a small job, then consider π' obtained from π by scheduling J_x in position v which gives $S_{\pi'(v-1)A}(\pi') = S_{\pi(v-1)A}(\pi) < S_{xA}(\pi^\star) \leq S_{\pi(v)A}(\pi) = S_{xA}(\pi')$. In this case J_x is a introductory job and verifies (iii). Let $\pi = \pi'$.

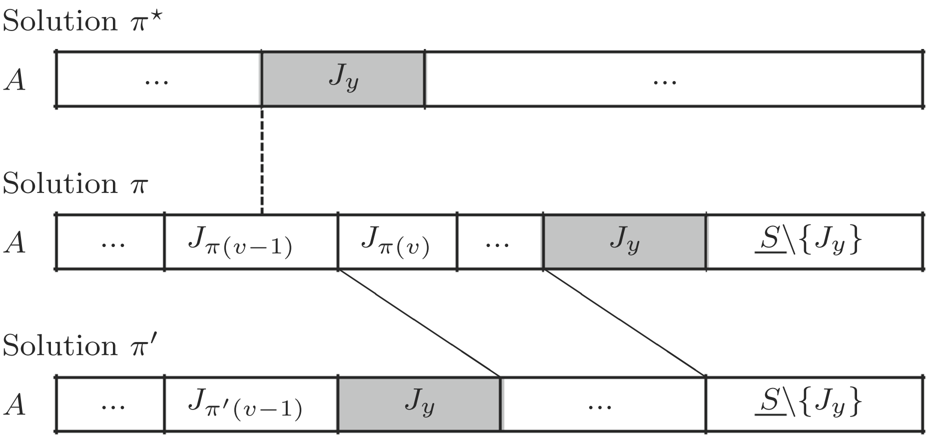

Fig. 2 Rescheduling the big jobs on π

By proceeding in the same way with the other big jobs in $\underline{S}\backslash\{J_x, J_y\}$, we will obtain a solution π generated by H^ε satisfying (iii) and the first inequality of (ii). We now prove that the generated solution π verifies the second inequality of (ii). Let $J_x = J_{\pi(w)}$ be a big job and let $J_y = J_{\pi(v)}$ be the introductory job of the block to which J_x belongs.

Note that $S_{xA}(\pi) = S_{yA}(\pi) + \sum_{i=v}^{w-1} a_{\pi(i)}$. Given that the big jobs have the same partial order in both π and $\pi^\star$ then $C_{xA}(\pi^\star) \geq S_{yA}(\pi^\star) + \sum_{i=v}^{w} a_{\pi(i)}$.

If J_y is first in π and $\pi^\star$ (i.e. $S_{yA}(\pi) = S_{yA}(\pi^\star) = 0$), then $C_{xA}(\pi^\star) \geq S_{xA}(\pi) + a_x > S_{xA}(\pi)$. Otherwise, from (iii) we obtain $S_{yA}(\pi^\star) > S_{\pi(v-1)A}(\pi) = S_{yA}(\pi) - a_{\pi(v-1)}$ which gives $C_{xA}(\pi^\star) \geq S_{xA}(\pi) + a_x - a_{\pi(v-1)} > S_{xA}(\pi)$ as $J_{\pi(v-1)}$ is a small job. □

Lemma 2

(i) *Let π be a schedule that verifies Lemma 1, and let $J_z = J_{\pi(u)}$ be the job which starts the last busy period on machine B in π. Given the job $J_{\pi^\star(u')}$ of $\pi^\star$ such that $S_{\pi^\star(u')A}(\pi^\star) \leq S_{zA}(\pi) < C_{\pi^\star(u')A}(\pi^\star)$, then*

$$\sum_{i=u}^{n} b_{\pi(i)} + H_{\pi(u)} \leq \sum_{i=u'}^{n} b_{\pi^\star(i)} + H_{\pi^\star(u')}.$$

(ii) *Suppose that $|\overline{J}| = 0$. Let $J_z = J_{\pi_0(u)}$ be the job which starts the last busy period on machine B in π_0, and let $J_{\pi^\star(u')}$ be the job of $\pi^\star$ such that $S_{\pi^\star(u')A}(\pi^\star) \leq S_{zA}(\pi_0) < C_{\pi^\star(u')A}(\pi^\star)$, then*

$$\sum_{i=u}^{n} b_{\pi_0(i)} + H_{\pi_0(u)} \leq \sum_{i=u'}^{n} b_{\pi^\star(i)} + H_{\pi^\star(u')}.$$

Proof (i) Let $E = \{J_{\pi(i)} \mid u \leq i \leq n\}$, $F = \{J_{\pi^\star(i)} \mid u' \leq i \leq n\}$ and $G = E \cap F$ (see Fig. 3). By assumption $S_{\pi^\star(u')A}(\pi^\star) \leq S_{zA}(\pi)$, hence $a(E) \leq a(F)$ and

$$a(E\backslash G) \leq a(F\backslash G). \tag{4}$$

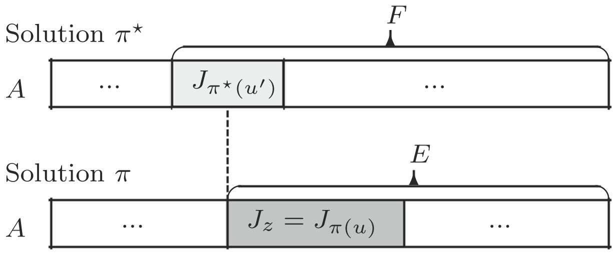

Fig. 3 Sets E and F

We first establish that $E \cap \overline{J} = F \cap \overline{J}$. For that the two following cases are considered:

Case 1. J_z is a small job.

Let $J_x \in F$ be a big job and suppose that $J_x \notin E$, hence $C_{xA}(\pi) \leq S_{zA}(\pi)$. From Lemma 1(ii) we have $S_{xA}(\pi^\star) \leq S_{xA}(\pi)$ and then $C_{xA}(\pi^\star) \leq C_{xA}(\pi)$. Consequently $C_{xA}(\pi^\star) \leq S_{zA}(\pi)$. As $J_x \in F$ then $C_{\pi^\star(u')A}(\pi^\star) \leq C_{xA}(\pi^\star)$, and consequently $C_{\pi^\star(u')A}(\pi^\star) \leq S_{zA}(\pi)$ which leads to a contradiction. Thus if $J_x \in F$ then $J_x \in E$.

Let $J_x \in E$ be a big job, and let $J_y = J_{\pi(v)}$ be the first introductory job scheduled after J_z in π. We have $S_{zA}(\pi) \leq S_{\pi(v-1)A}(\pi)$. Note that J_y cannot be the first job in $\pi^\star$ for otherwise J_y is also the first job in π which is not possible as J_y is scheduled after J_z and this latter is assumed to be a small job. From Lemma 1(iii) we have $S_{\pi(v-1)A}(\pi) < S_{yA}(\pi^\star)$ and consequently $S_{zA}(\pi) \leq S_{yA}(\pi^\star)$. By definition $S_{\pi^\star(u')A}(\pi^\star) \leq S_{zA}(\pi)$, hence $S_{\pi^\star(u')A}(\pi^\star) \leq S_{yA}(\pi^\star)$ and $J_y \in F$. Besides, the big jobs have the same order in π and $\pi^\star$, so, J_x is scheduled after J_y in $\pi^\star$ and consequently $J_x \in F$. therefore if $J_x \in E$ then $J_x \in F$.

Case 2. J_z is a big job.

Note that given Lemma 1(ii), $J_{\pi^\star(u')} = J_z$. Knowing that the big jobs appear in the same partial order in both π and $\pi^\star$ then $E \cap \overline{J} = F \cap \overline{J}$.

Considering the previous two cases, we conclude that $E \cap \overline{J} = F \cap \overline{J}$. As the jobs in π, except those of $\overline{J}$, are scheduled according to *RR*, and considering $J_{z'}$ the first job of $E \backslash G$, we have $b_{z'}/a_{z'} \geq b_i/a_i \ \forall J_i \in E \backslash G$, and $b_{z'}/a_{z'} \leq b_i/a_i \ \forall J_i \in F \backslash G$. Using (4), we derive that

$$\begin{aligned} b(E\backslash G) &\leq \sum_{J_i \in E\backslash G} a_i \left(\frac{b_i}{a_i}\right) \\ &\leq \sum_{J_i \in E\backslash G} a_i \left(\frac{b_{z'}}{a_{z'}}\right) \\ &\leq \sum_{J_i \in F\backslash G} a_i \left(\frac{b_{z'}}{a_{z'}}\right) \\ &\leq \sum_{J_i \in F\backslash G} a_i \left(\frac{b_i}{a_i}\right) \leq b(F\backslash G), \end{aligned}$$

and so

$$\sum_{i=u}^{n} b_{\pi(i)} \leq \sum_{i=u'}^{n} b_{\pi^\star(i)}. \tag{5}$$

As J_z starts a busy period on machine B, then (5) implies that $S_{\pi^\star(u')B}(\pi^\star) \leq S_{zB}(\pi)$, for otherwise $\pi^\star$ is not optimal. Hence $H_{\pi(u)} \leq H_{\pi^\star(u')}$ and

$$\sum_{i=u}^{n} b_{\pi(i)} + H_{\pi(u)} \leq \sum_{i=u'}^{n} b_{\pi^\star(i)} + H_{\pi^\star(u')}.$$

(ii) Given that all jobs in π_0 are scheduled according to *RR*, and using a similar argument as in (i), it should be easy to show the result. □

The worst-case performance of Algorithm H^ε is given by Theorem 1.

Theorem 1 *For the $F2|h(0,q), r, s_q < C^\star_{\max}|C_{\max}$ problem and a given $\varepsilon > 0$, the relative worst-case error bound of Algorithm H^ε is given by $C_H/C^\star_{\max} \leq (1+\varepsilon)$.*

Proof As explained in the proof of Lemma 1, $0 \leq |\overline{J}| \leq \lceil \frac{1}{\varepsilon} \rceil$. Two cases have to be discussed.

Case 1: $|\overline{J}| = 0$.

Consider schedule π_0 and let $J_z = J_{\pi_0(u)}$ be the job which starts the last busy period on machine B. Let $J_{\pi^\star(u')}$ be the job in $\pi^\star$ such that $S_{\pi^\star(u')A}(\pi^\star) \leq S_{zA}(\pi_0) < C_{\pi^\star(u')A}(\pi^\star)$.

If J_z satisfies (Condition 1), then (2) and Lemma 2(ii) imply that

$$\begin{aligned} C_{\max}(\pi_0) &= C_{zA}(\pi_0) + \sum_{i=u}^{n} b_{\pi_0(i)} + H_{\pi_0(u)} \\ &\leq S_{zA}(\pi_0) + a_z + \sum_{i=u'}^{n} b_{\pi^\star(i)} + H_{\pi^\star(u')} \\ &\leq C_{\pi^\star(u')A}(\pi^\star) + a_z + \sum_{i=u'}^{n} b_{\pi^\star(i)} + H_{\pi^\star(u')}. \end{aligned} \tag{6}$$

Given the position of $J_{\pi^\star(u')}$ in $\pi^\star$, we have $C_{\pi^\star(u')A}(\pi^\star) + \sum_{i=u'}^{n} b_{\pi^\star(i)} + H_{\pi^\star(u')} \leq C^\star_{\max}$. Hence, (6) gives $C_{\max}(\pi_0) \leq C^\star_{\max} + a_z \leq (1+\varepsilon)C^\star_{\max}$ as by assumption $a_z < \varepsilon C^\star_{\max}$.

If J_z satisfies (Condition 2), two cases have to be considered (see Fig. 4).

Case 1-1: $S_{\pi^\star(u')B}(\pi^\star) \geq t_r$. In this case $t_r + \sum_{i=u'}^{n} b_{\pi^\star(i)} + H_{\pi^\star(u')} \leq C^\star_{\max}$. Using (3) and Lemma 2(ii), we obtain $C_{\max}(\pi_0) = t_r + \sum_{i=u}^{n} b_{\pi_0(i)} + H_{\pi_0(u)} \leq C^\star_{\max}$.

Case 1-2: $S_{\pi^\star(u')B}(\pi^\star) < s_r$. Let $\delta = s_r - S_{\pi^\star(u')B}(\pi^\star)$. Note that by construction $\delta \leq a_z \leq \varepsilon C^\star_{\max}$ and that both π_0 and $\pi^\star$ are affected by the same holes. Hence, and given the position of $J_{\pi^\star(u')}$ in $\pi^\star$, we have $t_r + \sum_{i=u'}^{n} b_{\pi^\star(i)} + H_{\pi_0(u)} - \delta \le C^\star_{\max}$. Consequently, and using (3),

$$\begin{aligned} C_{\max}(\pi_0) &= t_r + \sum_{i=u}^{n} b_{\pi_0(i)} + H_{\pi_0(u)} \\ &\le t_r + \sum_{i=u'}^{n} b_{\pi^\star(i)} + H_{\pi_0(u)} \\ &\le C^\star_{\max} + \delta \le (1+\varepsilon)C^\star_{\max}. \end{aligned} \tag{7}$$

Case 2: $|\overline{J}| \neq 0$.

Consider a permutation π verifying Lemma 1. Let $J_z = J_{\pi(u)}$ be the job which starts the last busy period on machine B, and let $J_{\pi^\star(u')}$ be such that $S_{\pi^\star(u')A}(\pi^\star) \le S_{zA}(\pi) < C_{\pi^\star(u')A}(\pi^\star)$. The following two sub-cases are considered.

Case 2-1: $J_z \notin \overline{J}$.

Using exactly the same argument as in case 1, we derive $C_{\max}(\pi) \le C^\star_{\max} + a_z \le (1+\varepsilon)C^\star_{\max}$.

Case 2-2: $J_z \in \overline{J}$.

Let $J_y = J_{\pi(v)}$ be the introductory job of the block to which J_z belongs. Suppose that J_z follows (Condition 1). If J_y is the first job in π then it is also first in $\pi^\star$ (i.e. $v = 1$). Recall that all the big jobs scheduled between J_y and J_z in π are scheduled in the same order in $\pi^\star$. Thus, and given the position of J_y in $\pi^\star$, we have $\sum_{i=1}^{u} a_{\pi(i)} + \sum_{i=u'}^{n} b_{\pi^\star(i)} + H_{\pi^\star(u')} \le C^\star_{\max}$. Then (2) and Lemma 2(i) imply that

$$\begin{aligned} C_{\max}(\pi) &= C_{zA}(\pi) + \sum_{i=u}^{n} b_{\pi(i)} + H_{\pi(u)} \\ &\le \sum_{i=1}^{u} a_{\pi(i)} + \sum_{i=u'}^{n} b_{\pi^\star(i)} + H_{\pi^\star(u')} \le C^\star_{\max}. \end{aligned} \tag{8}$$

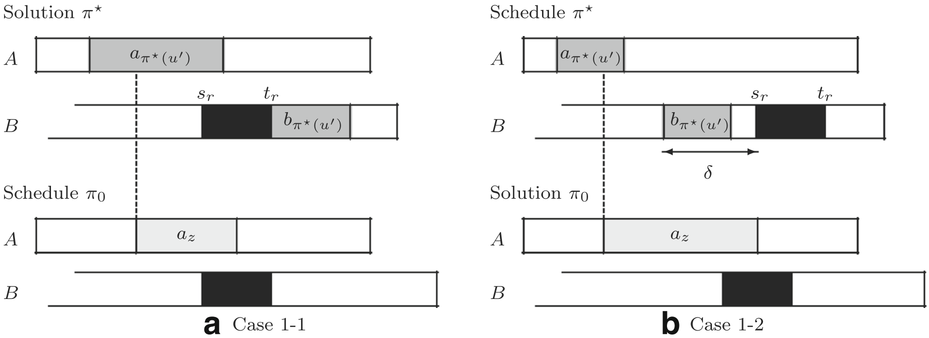

Fig. 4 Solution π_0 and $\pi^\star$

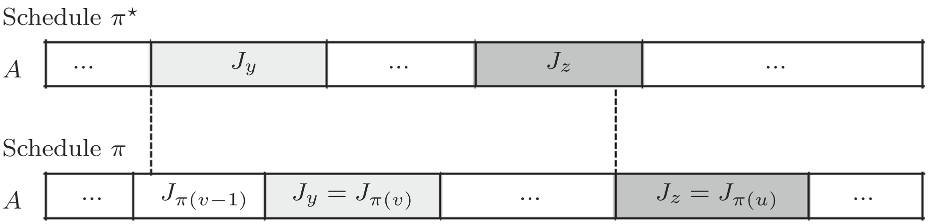

Fig. 5 Schedule π satisfying condition of case 2-2-1

If J_y is not first in π then it is preceded by the small job $J_{\pi(v-1)}$ (see Fig. 5). Using (2) and Lemma 2(i) we derive

$$\begin{aligned}C_{\max}(\pi) &= C_{zA}(\pi) + \sum_{i=u}^{n} b_{\pi(i)} + H_{\pi(u)} \\ &\leq S_{\pi(v-1)A}(\pi) + a_{\pi(v-1)} + \sum_{i=v}^{u} a_{\pi(i)} \\ &\quad + \sum_{i=u'}^{n} b_{\pi^\star(i)} + H_{\pi^\star(u')}. \end{aligned} \tag{9}$$

Recall that all the big jobs scheduled between J_y and J_z in π are scheduled in the same order in $\pi^\star$. Thus, and given the position of J_y in $\pi^\star$ we have $S_{yA}(\pi^\star) + \sum_{i=v}^{u} a_{\pi(i)} + \sum_{i=u'}^{n} b_{\pi^\star(i)} + H_{\pi^\star(u')} \leq C^\star_{\max}$. As J_y is an introductory job, then Lemma 1(iii) implies that $S_{\pi(v-1)A}(\pi) < S_{yA}(\pi^\star)$. Consequently (9) gives

$$C_{\max}(\pi) \leq C^\star_{\max} + a_{\pi(v-1)} \leq (1+\varepsilon)C^\star_{\max}. \tag{10}$$

If J_z follows (Condition 2), then as in Case 1, two configurations are to be considered:

Case 2-2-1: $S_{\pi^\star(u')B}(\pi^\star) \geq t_r$. As in Case 1-1, $t_r + \sum_{i=u'}^{n} b_{\pi^\star(i)} + H_{\pi^\star(u')} \leq C^\star_{\max}$. Using (3) and Lemma 2(i) we get $C_{\max}(\pi) = t_r + \sum_{i=u}^{n} b_{\pi(i)} + H_{\pi(u)} \leq C^\star_{\max}$.

Case 2-2-2: $S_{\pi^\star(u')B}(\pi^\star) < s_r$. Let $\delta = s_r - S_{\pi^\star(u')B}(\pi^\star)$.

Note that given Lemma 1(ii) then necessarily $J_{\pi^\star(u')} = J_z$. Knowing that all the big jobs scheduled between J_y and J_z in π appear in the same order in $\pi^\star$; then $S_{zA}(\pi) - S_{yA}(\pi) \leq S_{zA}(\pi^\star) - S_{yA}(\pi^\star)$ and

$$\begin{aligned}\delta &= s_r - S_{zB}(\pi^\star) \\ &\leq C_{zA}(\pi) - C_{zA}(\pi^\star) \\ &\leq S_{zA}(\pi) - S_{zA}(\pi^\star) \leq S_{yA}(\pi) - S_{yA}(\pi^\star).\end{aligned} \tag{11}$$

If J_y is first in π and $\pi^\star$ then $\delta = 0$, otherwise $\delta \leq a_{\pi(v-1)} \leq \varepsilon C^\star_{\max}$. Note that similar to Case 1-2, $t_r + \sum_{i=u'}^{n} b_{\pi^\star(i)} + H_{\pi(u)} - \delta \leq C^\star_{\max}$. Hence, and using (3), we obtain

$$\begin{aligned} C_{\max}(\pi) &= t_r + \sum_{i=u}^{n} b_{\pi(i)} + H_{\pi(u)} \\ &\leq t_r + \sum_{i=u'}^{n} b_{\pi^\star(i)} + H_{\pi(u)} \\ &\leq C^\star_{\max} + \delta \leq (1+\varepsilon) C^\star_{\max}. \end{aligned} \tag{12}$$

Considering all previous cases, we conclude that $C_H \leq (1+\varepsilon)C^\star_{\max}$. □

4 Conclusion

In this paper, we presented a polynomial-time approximation scheme for a particular case of the two-machine flow shop problem with several availability constraints on the second machine. An interesting issue that deserves future investigation is the extension of the obtained result to some particular configurations of the two-machine job shop problem.

References

1. Allaoui, H., Artiba, A.: Scheduling two-stage hybrid flow shop with availability constraints. Comput. Oper. Res. **33**, 1399–1419 (2006)
2. Garey, M.R., Johnson, D.S., Sethi, R.: The complexity of flow shop and job shop scheduling. Math. Oper. Res. **1**, 117–129 (1976)
3. Garey, M.R., Johnson, D.S.: Computers and Intractability. Freeman, San Francisco (1979)
4. Hadda, H.: A $\left(\frac{4}{3}\right)$-approximation algorithm for a special case of the two machine flow shop problem with several availability constraints. Optim. Lett. **3**, 583–592 (2009)
5. Hadda, H.: An improved algorithm for the two machine flow shop problem with several availability constraints. 4OR-Q. J. Oper. Res. **8**, 271–280 (2010)
6. Hadda, H.: A polynomial-time approximation scheme for the two machine flow shop problem with several availability constraints. Optim. Lett. **6**, 559–569 (2011)
7. Hadda, H., Dridi, N., Hajri-Gabouj, S.: An improved heuristic for two-machine flow shop scheduling with an avilability constraint and nonresumable jobs. 4OR-Q. J. Oper. Res. **8**, 87–99 (2010)
8. Hadda, H., Dridi, N., Hajri-Gabouj, S.: The two-stage assembly flow shop scheduling with an availability constraint: worst case analysis. J. Math. Model. Algoritm. Oper. Res. (2013). doi:10.1007/s10852-013-9235-7
9. Hoogeveen, J.A., Lenstra, J.K., Veltman, B.: Preemptive scheduling in a two-stage multiprocessor flow shop is NP-hard. Eur. J. Oper. Res. **89**, 172–175 (1996)
10. Johnson, S.M.: Optimal two- and three-stage production schedules with setup times included. Res. Log. Q. **1**, 61–68 (1954)
11. Kubzin, M.A., Potts, C.N., Strusevich, V.A.: Approximation results for flow shop scheduling problems with machine availability constraints. Comput. Oper. Res. **36**, 379–390 (2009)
12. Kubiak, W., Blazewicz, J., Formanowicz, P., Breit, J., Schmidt, G.: Two-machine flow shops with limited machine availability. Eur. J. Oper. Res. **136**, 528–540 (2002)
13. Lee, C.Y.: Minimizing the makespan in the two-machine flow shop scheduling problem with an availability constraint. Oper. Res. Lett. **20**, 129–139 (1997)
14. Lee, C.Y.: Two-machine flowshop scheduling with availability constraints. Eur. J. Oper. Res. **114**, 420–429 (1999)

15. Ng, C.T., Kovalyov, M.Y.: An FPTAS for scheduling a two-machine flowshop with one unavailability interval. Nav. Res. Log. **51**, 307–315 (2004)
16. Schuurman, P., Woeginger, G.J.: A polynomial time approximation scheme for the two-stage multiprocessor flow shop problem. Theor. Comput. Sci. **237**, 105–122 (2000)
17. Williamson, D.P., Hall, L.A., Hoogeveen, J.A., Hurkens, C.A.J., Lenstra, J.K., Sevast'janov, S.V., Shmoys, D.B.: Short shop schedules. Oper. Res. **45**, 288–294