有使用限制的二台机器流水作业问题 *

杨 名 [1] 鲁习文 [1†]

摘要 研究有使用限制的二台机器流水作业排序问题，目标为最小化最大完工时间，工件加工可以被机器的不可用时间段中断. 讨论两台机器上均有使用限制离线问题的可近似情形，并给出性能比为 $\frac{3}{2}$ 的近似算法. 同时还考虑在第二台机器上存在一个不可用时间段情况下的半在线问题，给出一个竞争比为 $\frac{3}{2}$ 的半在线算法.

关键词 排序，流水作业，使用限制，近似算法，竞争比

中图分类号 O223

数学分类号 68B20, 90B35

Two-Machine Flow Shop Problems with Availability Constraints

YANG Ming[1] LU Xiwen[1†]

Abstract This paper investigates the problems for two-machine flow shop scheduling with availability constraints. A resumable scenario is assumed, i.e., if a job cannot be finished before the interval it is continued after the machine becomes available again. The objective is to minimize the makespan. This paper first considers an approximate case of the problem with several availability constraints on both machines, presents an algorithm with performance ratio of $\frac{3}{2}$, then gives an algorithm with competitive ratio of $\frac{3}{2}$ for the semi-online problem with an availability constraint on the second machine.

Keywords scheduling, flow shop, availability constraint, approximation algorithm, competitive ratio

Chinese Library Classification O223

2010 Mathematics Subject Classification 68B20, 90B35

0 引 言

经典的排序问题中通常假设机器在加工工件的过程中是始终可用的，但这个假设在某些问题中并不成立. 在实际生产中，经常会因为生产计划的变更而导致机器在某个时间段不能使用或不能执行当前的加工任务. 最常见的情况是机器因为使用时间的限制需要定期进行检修，由于在检修期间机器不能执行加工任务，使得工件不能连续地进行加工. 因此，针对这种实际情况，研究者提出了一类新的排序问题模型，即机器有使用限

收稿日期：2011 年 2 月 8 日.

* 基金项目：上海自然科学基金 (09ZR1407200), 国家自然科学基金 (11071072)

1. 华东理工大学理学院数学系, 上海 200237; Department of Mathematics, East China University of Science and Technology, Shanghai 200237, China.

† 通讯作者 Corresponding author

制的排序问题. 经过近二十年的研究, 很多机器有使用限制的排序问题都得到了解决, 相关的结果可以参见文献 [1].

在经典的流水作业排序问题里, 给定二台机器 A 和 B, 以及 n 个工件 $J=(J_1,\cdots,J_n)$, 每个工件 J_j 都有两道工序, 需要依次在机器 A 和机器 B 上加工, 两道工序加工时间分别为 a_j 和 b_j, 工件表示为 $J_j=(a_j,b_j)$. 每个工件一次只能被一台机器加工, 每台机器一次最多只能加工一个工件的一道工序. 目标为最小化最大完工时间, 即最小化最后工件的完工时间. 在本文的模型中, 机器并不一直可用, 加工时间表中存在多个不可用时间段, 通常我们称这些不可用时间段为 "洞". 根据 Lee[2] 的分类, 可以将工件被不可使用时间段中断时的情况分为三种类型: 可中断 (resumable) 的情况, 即若某工件在洞之前没有完工, 则可以在洞结束后对其剩余的部分继续进行加工; 工件半中断 (semi-resumable) 的情况, 即中断工件在恢复加工时, 该工件已被加工的部分中有一部分要重新加工; 不可以中断 (non-resumable) 的情况, 即若工件在洞之前未能完成加工, 则必须在洞结束之后重新对其进行加工. 该问题用三参数表示法可表示为 $F2|h(q_A,q_B),S_C|C_{\max}$, 其中 q_A, q_B 分别表示第一和第二台机器上洞的个数, $S_C\in\{Re,S-Re,N-Re\}$ 分别表示上述三种类型的情况 [3].

Johnson 证明了的 $F2||C_{\max}$ 问题是可解的, 并给出了求解该问题的 Johnson 算法 [4]. 如果机器存在不可使用的时间段时, 问题将会变得更难. Lee 研究了在机器只存在一个洞的情况下, 对于上述三种类型问题均是 NP- 难的 [2,5], 他同时给出了拟多项式动态规划算法. 若有多个洞在某一台机器上, Kubiak 等人证明了在工件可中断的情况下, 问题为强 NP- 难的 [6]. 而对于半中断和不可中断的情况, Lee 证明了单机有多洞的问题也为强 NP- 难的 [7]. 作为更一般的情况, 相应流水作业问题也为强 NP- 难的.

对可中断的情况, 当第二台机器上存在两个洞或两台机器各有一个洞的时候, 问题不存在常数性能比的近似算法 [2,6]. Breit 证明了对于半中断和不可中断的情况 [8], 单机有两个洞的排序问题不存在常数性能比的近似算法. 因此对于两台机器流水作业问题, 只有在机器存在一个洞 (对三种情况均成立) 和可中断情况下第一台机器存在多个洞的时候, 问题才存在常数性能比的近似算法 [3].

在工件可中断的情况下, Lee 对 $F2|h(1,0),Re|C_{\max}$ 问题给出了性能比为 $\frac{3}{2}$ 的近似算法 [5], 对 $F2|h(0,1),Re|C_{\max}$ 给出了 $\frac{4}{3}$ 近似的算法. 对前一个问题 Cheng 和 Wang[9] 将性能比改进到 $\frac{4}{3}$, 后一个问题 Breit 给出了一个 $\frac{5}{4}$ 近似的算法 [10]. 当机器只存在一个洞的情况下, 问题有 PTAS[11] 和 FPTAS[12]. 若第一台机器上有多个洞, 即 $F2|h(q,0),Re|C_{\max}$, Kubzin 首先给出了一个 $\frac{3}{2}$ 近似的算法 [3], Hadda[13] 则将其改进到 $\frac{4}{3}$. 若第二台机器上有多个洞, 但最后一个洞的结束时间有限制, 对于这个问题, Hadda 给出了一个 4/3 近似的算法 [14].

在工件半中断或不可中断的情况下, Lee 首先对 $F2|h(1,0),S-Re|C_{\max}$ 给出了 2 近似的算法 [2], 对 $F2|h(0,1),S-Re|C_{\max}$ 给出了 $\frac{3}{2}$ 近似的算法. 对前一个问题, Hadda 给出了一个 3/2 近似的算法 [15]. 针对两台机器上各有一个洞, 且洞是连续, 即一个洞的结束时间为另一个洞的开始时间的问题, Cheng 和 Wang 给出了一个 $\frac{5}{3}$ 的近似算法 [16].

本文研究了两台机器上均存在多个洞的情况下的一个可近似问题, 证明该问题具有性能比为 $\frac{3}{2}$ 的近似算法. 而关于在线的机器有使用限制的流水作业问题, 目前还没有人进行过研究. 本文第二部分, 将研究第二台机器上有一个洞的半在线情况, 给出一个竞争比为 $\frac{3}{2}$ 的在线算法.

1 问题及符号描述

由 Kubzin 等人 [3] 的文章可以得知，在两台机器上如果都存在多个洞的情况下，问题没有常数性能比的近似算法。因此我们考虑该离线问题的一种可近似情况：两台机器均存在多洞并且对第二台机器上最后一个洞的结束时间加以限制．首先对一些符号进行定义：

$q_A(q_B)$ ：机器 A（ B ）上洞的个数.

$H_A(t)(H_B(t))$ ： t 时刻后机器 A（ B ）上洞长之和.

$C_{\max}(\sigma)$ ：算法得到的时间表 σ 的表长.

$C^*_{\max}$ ：最优时间表 π 的时间表长.

C^J ： $F2||C_{\max}$ 问题的最优表长.

$a(S)(b(S))$: 工件集合 S 中第一（二）道工序之和.

$S_i^A(S_i^B)$ ：工件 J_i 在机器 A（ B ）上的开工时间.

$C_i^A(C_i^B)$ ：工件 J_i 在机器 A（ B ）上的完工时间.

与 [14] 里的假设一样，令机器 B 上最后一个洞的结束时间 t_{q_B} 小于 C^J, 记该问题为 $F2|h(q_A,q_B),Re,t_{q_B}<C^J|C_{\max}$. 尽管两台机器均有多洞的情形是不可近似的，在下一节我们将证明该问题是可近似的．对于这种可近似情况，显然有 $C^J\leqslant C^*_{\max}$. 且在这种特殊情况下，机器 B 上的洞会影响所有的时间表．因此最优解的下界有 $C^*_{\max}\geqslant b(J)+H_B(0)$.

而对于半在线问题，我们主要研究 $F2|h(0,1),Re|C_{\max}$ 问题，其中工件是逐个到达的 (over list), 最优时间表长 $C^*_{\max}$ 已知，且洞的结束时间不超过 $\frac{1}{2}C^*_{\max}$.

根据 Kubzin 等人 [3] 的结果，我们在设计算法时只需考虑问题的排列排序时间表，即在所得的时间表里，两台机器加工工件的次序一样．同时算法在执行时，工件的工序总是尽可能早的加工.

2 离线问题

现在，我们对问题 $F2|h(q_A,q_B),Re,t_{q_B}<C^J|C_{\max}$ 进行讨论，将 Kubzin 等人的算法应用于我们所讨论的问题里，并且证明该算法的性能比为 $\frac{3}{2}$.

算法 H1

步骤 1 将第二道工序加工时间最大的工件放在时间表的首位，剩余的工件任意排列，记这个时间表为 σ_1;

步骤 2 将所有工件按 $\frac{b_i}{a_i}$ 非增的顺序加工，记这个时间表为 σ_2;

步骤 3 选择 σ_1 和 σ_2 中时间表长最短的一个时间表作为算法的输出时间表.

在分析算法性能比之前，给出一个引理．首先我们定义这里的关键工件 J_u 为时间表 σ 中机器 B 上最后一段忙碌时间段的开始工件，即从 b_u 工序开始直至所有工件完工的这段时间段里，机器 B 不存在空闲，且 b_u 工序之前是机器空闲或是机器的不可用时间段.

引理 1 如果 J_u 是算法产生的时间表 σ_2 的关键工件，则有

$$C_{\max}(\sigma_2)-C^*_{\max}\leqslant b_u.$$

证明 不妨设时间表 σ_2 中的关键工件 J_u 在机器 A 上的完工时间为 t. 令最优时间表 π 中第一道工序完工时间不早于 t 的工件集合为 S_1, σ_2 中排在 J_u 之后的工件集合为 S_2.

则

$$a(S_2)+H_A(t)<a(S_1)+H_A(t).$$

令 $S_3=S_1\bigcap S_2$, 有

$$a(S_2\backslash S_3)\leqslant a(S_1\backslash S_3),$$

且对 $\forall J_i\in S_1\backslash S_3$ ，有

$$\frac{b_i}{a_i}\geqslant\frac{b_u}{a_u},$$

以及 $\forall J_i\in S_2\backslash S_3$ ，有

$$\frac{b_i}{a_i}\leqslant\frac{b_u}{a_u}.$$

则

$$\begin{aligned}\sum_{J_i\in S_2\backslash S_3}b_i&=\sum_{J_i\in S_2\backslash S_3}a_i\Big(\frac{b_i}{a_i}\Big)\leqslant\sum_{J_i\in S_2\backslash S_3}a_i\Big(\frac{b_u}{a_u}\Big)\\&\leqslant\sum_{J_i\in S_1\backslash S_3}a_i\Big(\frac{b_i}{a_i}\Big)\leqslant\sum_{J_i\in S_1\backslash S_3}a_i\Big(\frac{b_i}{a_i}\Big)=\sum_{J_i\in S_1\backslash S_3}b_i\end{aligned}$$

因此有 $b(S_2)\leqslant b(S_1)$, 注意到在最优时间表 π 里，工件集 S_1 里工件的第二道工序均在时刻 t 之后完工,

$$C^*_{\max}\geqslant t+b(S_1)+H_B(t).$$

这样若关键工件 J_u 的第二道工序之前是空闲时间段，有

$$\begin{aligned}C_{\max}(\sigma_2)&=t+b_u+b(S_2)+H_B(t)\\&\leqslant t+b(S_1)+b_u+H_B(t)\\&\leqslant C^*_{\max}+b_u.\end{aligned}$$

若 J_u 的第二道工序之前是不可用时间段，则不妨令该不可用时间段的结束时间为 t_k ，同理有

$$\begin{aligned}C^*_{\max}&\geqslant t_k+b(S_1)+H_B(t_k),\\C_{\max}(\sigma_2)&\leqslant C^*_{\max}+b_u.\end{aligned}$$

证毕.

下面我们分析算法的性能比.

定理 1 算法 H1 对问题 $F2|h(q_A,q_B),Re,t_{q_B}<C^J|C_{\max}$ 的性能比是 $\frac{3}{2}$.

证明 情形 1 ：工件集中不存在第二道工序加工时间大于 $\frac{1}{2}C^*_{\max}$ 的工件，即 $\forall i=1,\cdots,n$ ，有 $b_i\leqslant\frac{1}{2}C^*_{\max}$. 我们考虑算法中的时间表 σ_2. 由引理 1 ，有

$$C_{\max}(\sigma_2)-C^*_{\max}\leqslant b_u,$$

显然

$$b_u\leqslant\frac{1}{2}C^*_{\max}.$$

因此

$$C_{\max}(\sigma_2)-C^*_{\max}\leqslant\frac{1}{2}C^*_{\max}.$$

情形 2：存在第二道工序加工时间大于 $\frac{1}{2}C^*_{\max}$ 的工件，显然只有一个这种工件．由算法 H 中的时间表 σ_1，该工件放在首位，不妨记该工件为 J_1．则

$$\sum_{i=2}^{n} b_i + H_B(0) < \frac{1}{2}C^*_{\max}.$$

若 J_1 是关键工件．最优时间解中会加工 J_1, 因此 J_1 在 σ_1 中的完工时间是最优值的一个下界．则

$$\begin{aligned} C_{\max}(\sigma_1) &\leqslant C^*_{\max} + H_B(0) + \sum_{i=2}^{n} b_i \\ &\leqslant C^*_{\max} + \frac{1}{2}C^*_{\max} = \frac{3}{2}C^*_{\max}. \end{aligned}$$

若 J_1 不是关键工件，在这种情形下关键工件在机器 A 上的完工时间也是最优时间表长的一个下界．而由上述的推导，无论关键工件第二道工序之前是空闲时间或是不可用时间段，可以得到

$$C_{\max}(\sigma_1) \leqslant C^*_{\max} + \frac{1}{2}C^*_{\max} = \frac{3}{2}C^*_{\max}.$$

由于该问题是 [3] 中问题的更一般情况，因此，算法的紧性依然成立．

3 半在线问题

至今还没有人对在线机器有使用限制的流水作业问题进行研究．因此，在本节我们将对 $F2|h(0,1), Re|C_{\max}$ 问题的一种半在线情况给出一个竞争比为 $\frac{3}{2}$ 的算法．工件是逐个到达的，最优时间表长 $C^*_{\max}$ 已知且洞的结束时间不超过 $\frac{1}{2}C^*_{\max}$．由于我们已经知道最优时间表长 $C^*_{\max}$, 因此在构造算法时，会事先安排好一段时间段 $[0, \frac{3}{2}C^*_{\max}]$ 来放置工件．工件在到达后，我们会按照算法来安排工件．最后将所有工件都放入到这段时间段中，我们会证明这样安排工件不会导致工件工序冲突．

以各自的到达顺序，分别记 $a_i \leqslant b_i$ 的工件为 J_i, $a_i > b_i$ 的工件为 $J_{i'}$．

算法 H2 每当有工件 $J_i(J_{i'})$ 出现：

步骤 1 若工件是 J_i, 则安排工件，使得

$$S^A_i = C^A_{i-1} \quad \text{且} \quad S^B_i = \max\{C^B_{i-1}, C^A_i\}.$$

若 J_1 是第一个这样的工件，则令 $S^A_1 = 0$ 且 $S^B_1 = C^A_1$.

步骤 2 若工件是 $J_{i'}$, 则安排工件使之满足 $C^B_{i'} = S^B_{(i-1)'}$ 且 $C^A_{i'} = \min\{S^A_{(i-1)'}, S^B_{i'}\}$. 对于第一个这种工件 $J_{1'}$, 令 $C^B_{1'} = \frac{3}{2}C^*_{\max}$ 且 $C^A_{1'} = S^B_{1'}$.

步骤 3 若在安排工件时遇到洞，则中断工件，将剩余的部分安排在洞之后．

下面我们来证明算法对于问题的竞争比是 $\frac{3}{2}$.

定理 2 算法 H2 的竞争比是 $\frac{3}{2}$.

证明 显然，如果工件都能被安排到我们之前规划好的时间段 $[0, \frac{3}{2}C^*_{\max}]$ 内，且工序之间不会产生冲突，则有

$$C^{H2}_{\max} \leqslant \frac{3}{2}C^*_{\max}.$$

根据算法，安排工件的时候都满足了工序之间的加工次序．因此，这里我们只需证明所有工件都能被安排入时间段，而不会产生剩余的时间段小于出现的工件工序的加工时间这种情况即可．同时注意到，根据算法，所有 $a_i \leqslant b_i$ 的工件的第一道工序在机器 A 上是连续加工的，所有 $a_i > b_i$ 的工件的第二道工序在机器 B 上也是连续加工的．

采用反证法，存在工件无法被放入时间段中．假设算法 H2 已经往时间段里安排了 k 个工件，而第 $k+1$ 个工件 J_{k+1} 无法放入到时间段中．这里为了证明方便，我们将已经安排好的工件按它们的开工时间重新给它们标号，且记洞的长度为 H．

显然最优时间表长有

$$C^*_{\max} \geqslant \sum_{i=1}^{k} b_i + b_{k+1} + H, \quad C^*_{\max} \geqslant \sum_{i=1}^{k} a_i + a_{k+1}$$

和

$$C^*_{\max} \geqslant \max_{1\leqslant i\leqslant n} (a_i + b_i).$$

情形 1: 若 $a_{k+1} \leqslant b_{k+1}$．由算法可以得知，在安排工件时，先安排工件的第一道工序，然后才安排第二道工序．此时会出现两种情况：

情形 1.1: 按照算法，第一道工序无法放入时间段中．若记时间表中第一台机器上，算法用于安排工件的空闲时间段为 I_A ，则有 $a_{k+1} > I_A$．

此时根据算法，在算法生成的时间表中，至少存在一个工件 J_i 满足 $a_i > b_i$ ，且 $S_i^B = C_i^A$, 记关键工件 J_u 为其中最早开始加工的工件．这种情形下，安排在 J_u 之后的工件都满足第一道工序大于第二道工序．同时由洞的结束时间不超过 $\frac{1}{2}C^*_{\max}$ 的这个条件，我们知道洞的结束时间不会比在 J_u 的第二道工序开工时间晚．否则由算法的时间表构造可知，安排在洞之后的工件第二道工序之和将大于等于 $C^*_{\max}$, 与条件 $C^*_{\max} \geqslant \sum\limits_{i=1}^{n} b_i + H$ 矛盾．因此有

$$\sum_{i=1}^{u} a_i + I_A + \sum_{i=u}^{k} b_i = \frac{3}{2} C^*_{\max}.$$

而由假设，有

$$\sum_{i=1}^{u} a_i + a_{k+1} + \sum_{i=u}^{k} b_i > \frac{3}{2} C^*_{\max}.$$

利用上面得出的最优值的下界，可以得到：

$$\begin{aligned}
\frac{1}{2}C^*_{\max} &< \sum_{i=1}^{u} a_i + a_{k+1} + \sum_{i=u}^{k} b_i - C^*_{\max} \\
&\leqslant \sum_{i=1}^{u} a_i + a_{k+1} + \sum_{i=u}^{k} b_i - \left(\sum_{i=1}^{k} a_i + a_{k+1} \right) \\
&= b_u + \sum_{i=u+1}^{k} (b_i - a_i) \\
&\leqslant b_u \leqslant \frac{1}{2}(a_u + b_u) \leqslant \frac{1}{2} C^*_{\max}
\end{aligned}$$

矛盾，这说明假设不成立.

情形 1.2: 假设第一道工序已经放入时间段，但第二道工序无法放入时间段中. 记 ΔH 为洞影响时间表长的部分，显然 $0 \leqslant \Delta H \leqslant H$. J_v 为该时间表中满足 $a_v > b_v$ 的最早开工的工件，显然排在 J_v 之前的工件均满足第一道工序不超过第二道工序. 此时有

$$\sum_{i=1}^{v-1} a_i + a_{k+1} + b_{k+1} + \sum_{i=v}^{k} b_i + \Delta H > \frac{3}{2} C^*_{\max}.$$

我们有

$$\frac{1}{2} C^*_{\max} < \sum_{i=1}^{v-1} a_i + a_{k+1} + b_{k+1} + \sum_{i=v}^{k} b_i + \Delta H - \left(\sum_{i=1}^{k} b_i + b_{k+1} + H \right) \leqslant a_{k+1} \leqslant \frac{1}{2} C^*_{\max}.$$

显然这种情况也会产生矛盾.

情形 2: 若 $a_{k+1} > b_{k+1}$. 对于这种工件，根据算法，是先安排第二道工序，然后才安排第一道工序. 因此同情形 1 ，我们这里也分两种情况.

情形 2.1: J_{k+1} 的第二道工序无法放入时间段中. 根据算法生成的时间表，至少存在一个关键工件 J_i 满足 $a_i \leqslant b_i$, 且 $S_i^B = C_i^A$ 或者 $S_i^B = \Delta H + C_i^A$(即工件第一道工序结束时，第二台机器上可能存在洞，使得工件的第二道工序要等到洞结束时才能开始加工.) 我们记关键工件 J_u 为其中最晚开始加工的工件. 因此，有

$$\sum_{i=1}^{u} a_i + b_{k+1} + \sum_{i=u}^{k} b_i + \Delta H > \frac{3}{2} C^*_{\max}.$$

之后的分析同情形 1.2, 得到

$$\frac{1}{2} C^*_{\max} < \sum_{i=1}^{u} a_i + b_{k+1} + \sum_{i=u}^{k} b_i + \Delta H - C^*_{\max} \leqslant a_u \leqslant \frac{1}{2} C^*_{\max}$$

矛盾.

情形 2.2: J_{k+1} 的第二道工序放入时间段后，第一道工序无法放入时间段. 类似地记 J_v 为该时间表中满足 $a_v > b_v$ 的最早开工的工件. 此时有

$$\sum_{i=1}^{v-1} a_i + a_{k+1} + b_{k+1} + \sum_{i=v}^{k} b_i > \frac{3}{2} C^*_{\max}.$$

这种情况同情形 1.1 一样，因此也不可能出现.

综合上面的分析，我们证明了所有工件均可以放入时间段且工序之间不产生冲突，定理得证.

最后我们给出一个实例，证明算法是紧的，即算法的下界是 $\frac{3}{2}$.

事先已知最优时间表长 $C^*_{\max} = 2 + \varepsilon$, 洞的开始和结束时间均为 0. 第一个工件 $J_1 = (1 + \varepsilon, 1)$ 到达，然后是第二个工件 $J_2 = (1, 0)$. 根据算法 H2, 两个工件先后放入事先规定好的时间段 $[0, 3 + \frac{3}{2}\varepsilon]$ 中，

$$C^{H2}_{\max} = 3 + \frac{3}{2}\varepsilon.$$

令 ε 趋向于 0 ,

$$\frac{C_{\max}^{H2}}{C_{\max}^{*}} \to \frac{3}{2}.$$

参 考 文 献

[1] Ma Y, Chu C, Zuo C. A survey of scheduling with deterministic machine availability constraints[J]. *Computers and Industrial Engineering*, 2010, **58**: 199-211.

[2] Lee C Y. Two-machine flowshop scheduling with availability constraints[J]. *European Journal of Operational Research*, 1999, **114**: 420-429.

[3] Kubzin M A, Potts C N, Strusevich V A. Approximation results for flow shops scheduling problems with machine availability constraints[J]. *Computers and Operations Research*, 2009, **36**: 379-390.

[4] Johnson S M. Optimal two-and-three-stage production schedules with set-up times included[J]. *Naval Research Logistics Quart*, 1954, **1**: 61-68.

[5] Lee C Y. Minimizing the makespan in the two-machine flowshop scheduling problem with availability constraint[J]. *Operations Research Letters*, 1997, **20**: 129-139.

[6] Kubiak W, Blazewicz J, Formanowicz P, et al. Two-machine flow shops with limited machine availability[J]. *European Journal of Operational Research*, 2002, **136**: 528-540.

[7] Lee C Y. Machine scheduling with an availability constraint[J]. *Journal of Global Optimization*, 1996, **9**: 395-416.

[8] Breit J, Schmidt G, Strusevich V A. Non-preemptive two-machine open shop scheduling with non-availability constraints[J]. *Mathenatical Methods of Operations Research*, 2003, **57**: 217-234.

[9] Cheng T C E, Wang G. An improved heuristic for two-machine flowshop scheduling with an availability constraint[J]. *Operations Research Letters*, 2000, **26**: 223-239.

[10] Breit J. An improved approximation algorithm for two-machine flow shop scheduling with an availability constraint[J]. *Information Processing Letters*, 2004, **90**: 273-278.

[11] Breit J. A polynomial-time approximation scheme for the two-machine flow shop scheduling problem with an availability constraint[J]. *Computers and Operations Research*, 2006, **33**: 2143-2153.

[12] Ng C T, Kovalyov M Y. An FPTAS for scheduling a two-machine flowshop with one unavailability interval[J]. *Naval Research Logistics*, 2004, **51**: 307-315.

[13] Hadda H. An improved algorithm for the two machine flow shop problem with several availability constraints[J]. *4OR: A Quarterly Journal of Operations Research*, 2010, **8**: 271-280.

[14] Hadda H. A $\frac{4}{3}$-approximation algorithm for a special case of the two machine flow shop problem with several availability constraints[J]. *Optimization Letters*, 2009, **3**: 583-592.

[15] Hadda H, Dridi N, Hajri-Gabouj S. An improved heuristic for two-machine flow shop scheduling with an availability constraint and nonresumable jobs[J]. *4OR: A Quarterly Journal of Operations Research*, 2010, **8**: 87-99.

[16] Cheng T C E, Wang G. Two-machine flowshop scheduling with consecutive availability constraints[J]. *Information Processing Letters*, 1999, **71**: 49-54.