

The Two-Stage Assembly Flow Shop Scheduling with an Availability Constraint

Hatem Hadda

Unité de Recherche URAII, INSAT, Centre Urbain Nord B.P. N°676, 1080 Tunis, hatem.hadda@esti.rnu.tn

Najoua Dridi

Unité de Recherche OASIS, ENIT, B.P. N°37, le belvédère 100 Tunis, najoua.dridi@enit.rnu.tn

Sonia Hajri-Gabouj

Unité de Recherche URAII, INSAT, Centre Urbain Nord B.P. N°676, 1080 Tunis, sonia.gabouj@insat.rnu.tn

In this paper, we deal with the two-stage assembly flow shop scheduling problem with an availability constraint. The objective is to minimize the maximum completion time of the jobs (i.e. makespan). We are interested in the nonresumable model. We provide two heuristics with tight worst-case ratio bound of 2 when the availability constraint is imposed on one of the first stage's machines. For the case where the gap is imposed on the assembly machine we show that no polynomial heuristic algorithm with finite error bound can be found.

Keywords: Machine Scheduling, assembly flow shop, availability constraint, worst-case analysis.

1. Introduction

While most papers dealing with the scheduling field assume that the machines are available all the time, a growing number of works are considering that the machines are subject to breakdown; authors differentiate between the *probabilistic* case where breakdowns are random and the *deterministic* case where the date of unavailability period and its duration are known. Different models are distinguished, the *semiresumable* (*sr*) case that if a job cannot be finished before the next down period of a machine then the job will have to partially restart when the machine becomes available again. This model contains two special cases: namely the *resumable* (*r*) when the job can be continued without any penalty and the *nonresumable* (*nr*) case when the job needs to totally restart. The availability constraint was considered for many scheduling problems. The one machine problem was tackled by [1,6,17,18,21], the parallel machines by [13,19]. For the flow shop problem Lee [14] considered the two-machine flow shop problem with a resumable availability constraint in one of the two machines, he proved that the problem is NP-hard in the ordinary sense and developed a pseudo-polynomial dynamic programming algorithm to solve the problem. He also provided two heuristics with their error bound analysis. Later, Lee [15] studied the same problem but within the semiresumable case and availability constraints in both machines. He provided complexity analysis, developed a pseudo-polynomial dynamic programming and also proposed heuristic algorithms with an error bound analysis. Cheng and Wang [7] provided an improved approximation algorithm, for the case where the unavailability period is on the first machine; this algorithm has a relative worst-case error bound of $4/3$. Later, Breit [5] presented for the problem with one unavailability constraint on the second machine, an improved algorithm with a relative worst-case error bound of $5/4$. The two-machine flow shop problem with many unavailability periods was also studied by [4,12]. The general m machine flow shop problem was considered by Aggoune [2], for which he proposed a heuristic approach based on a genetic algorithm and a tabu search. More recently, Allaoui and Artiba [3] investigated the two-stage hybrid flow shop scheduling problem, with only one machine on the first stage and m machines on the second stage, the authors discussed the complexity of the problem, gave a branch and bound model and calculated the worst-case performances of three heuristics.

In this paper, we deal with the two-stage assembly flow shop scheduling with *nonresumable* availability constraint in one of the $m+1$ machines. The objective is to minimize the maximum comple-

tion time of the jobs, i.e., the makespan. We will use the notation $Am|nr - a(M_k)|C_{\max}$ to denote this problem, where M_k designates the machine subject to the availability constraint.

When considering the problem $Am||C_{\max}$ (without any machine availability constraint), we obtain the known Two-Stage Assembly Flow Shop problem. This problem can be viewed as an extension of the two-machine flow shop problem. The assembly problem was first introduced by Lee and al. [16] who studied the 3-Machines assembly flow shop (3MAF), where there is two machines in the first stage and a unique assembly machine in the second stage. The objective is to minimize the makespan. The problem was shown to be NP-hard in the strong sense and a number of polynomial cases was identified. Authors also proposed a branch and bound algorithm and three heuristics with their error bound analysis. Potts and al. [20] proposed a generalization to 3MAF problem by considering more than two machines at the first stage, which they noted $Am||C_{\max}$. The authors presented a complexity study and proposed a number of heuristics with a study of their relative and absolute errors.

Hariri and Potts [10] proposed a set of elimination rules which they used in a branch and bound algorithm, they were able to solve big size problems. More recently, Xi and al. [22] presented, for the 3MAF, a series of heuristics based on the ideas of Johnson [11] and Gupta [8], they also gave a worst-case analysis.

In addition, Haouari and Daouas [9] dealt with the inverse problem of 3MAF, they called this problem the 3-machine dismantling flow shop problem (3MDF) and they showed that this problem is equivalent to the original problem (3MAF). The authors also proposed an efficient branch and bound algorithm.

This paper is organized as follows. After presenting the notations and the problem description, we discuss in section 3 the configuration where the unavailability period is on one of the first stage's machines. We namely develop heuristic methods with an error bound analysis. In section 4, we investigate the case where the gap is on the assembly machine, a worst-case analysis is also provided.

2. Notations and problem description

For defining formally the problem, let us define:

n	total number of jobs to be scheduled;
J_i	job i , $i \in \{1, \dots, n\}$;
J	set of all the jobs: $J = \{J_1, J_2, \dots, J_n\}$;
M_A	assembly machine;
M_k	one of the first stage's machines, $k \in \{1, \dots, m\}$;
p_{ik}	processing time of job i on M_k , $k \in \{A, 1, \dots, m\}$;
$\pi = (\pi_1, \dots, \pi_n)$	a schedule of n jobs;
$C_{\max}(\pi)$	makespan of the schedule π ;
s_k	start time of the unavailability period on M_k , $k \in \{A, 1, \dots, m\}$;
t_k	finish time of the unavailability period on M_k , $k \in \{A, 1, \dots, m\}$;
C_{\max}^*	makespan of the optimal solution;
J_k^π	the crossover job of M_k under π .

The Two-Stage Assembly Flow Shop problem ($Am||C_{\max}$) is given by a set of n available jobs to be processed on a set of $(m+1)$ machines. Each job i consists of $m+1$ operations $\{O_{i1}, O_{i2}, \dots, O_{im}, O_{iA}\}$ to be processed respectively on M_k $k \in \{1, \dots, m\}$ and M_A . The operation O_{iA} cannot begin

until all operation O_{ik} $k \in \{1, \dots, m\}$ are finished. Besides, the n jobs are simultaneously available at time 0, each machine can handle no more than one job at a time.

We assume that one of the machines M_k $k \in \{A, 1, \dots, m\}$ is unavailable during the period from s_k to t_k ($0 \leq s_k \leq t_k$). If a job cannot be finished before the unavailability period (the gap) of a machine then the job has to restart. This case is known as *nonresumable*. Hence the problem will be denoted by $Am|nr - a(M_k)|C_{\max}$. The problem is to find the schedule that minimizes the total completion date of the last job visiting the machines M_A (i.e. the makespan $C_{\max}(\pi)$).

For a given schedule π , let J_k^π be the first job that finished after t_k on M_k . We call J_k^π the crossover job of M_k under π .

As the $Am||C_{\max}$ is NP-hard in the strong sense [16,20], then our more general problem $Am|nr - a(M_k)|C_{\max}$ is also strongly NP-hard. We note that the only variant of this problem solved in polynomial time is the well-known $F2||C_{\max}$ problem solved by Johnson [11].

3. Availability constraint on machine $M_h, 1 \leq h \leq m : Am|nr - a(M_h)|C_{\max}$

In this configuration, only the machine M_h for a fixed $1 \leq h \leq m$ is unavailable during the period from s_h to t_h .

3.1. Basic proprieties

We first point out that if $\sum_{1 \leq i \leq n} p_{ih} \leq s_h$, then the availability constraint does not affect the schedule, hence we assume that $\sum_{1 \leq i \leq n} p_{ih} > s_h$ and thus the completion time on M_h for any schedule will be greater than $\sum_{1 \leq i \leq n} p_{ih} + t_h - s_h$, hence:

$$\sum_{1 \leq i \leq n} p_{ih} + t_h - s_h \leq C_{\max}^* \quad (1)$$

For a given sequence of jobs π on M_h , there would be two sets of jobs, those which will be finished on M_h before the s_h (called the set X_π) and those which will be finished after the t_h (called the set Y_π). We have $\sum_{i \in X} p_{ih} \leq s_h$, we then note $\Delta_\pi = s_h - \sum_{i \in X} p_{ih}$. (We will drop the reference to the sequence π whenever no confusion can arise).

We also remark that for a given schedule π , and if we note $J_{i_0} = J_h^\pi$ then :

$$p_{i_0 h} > \Delta \quad (2)$$

Otherwise, J_{i_0} will be finished before s_h on M_h .

Note that it is sufficient to consider the *permutation solution* for $Am||C_{\max}$ [20]. However, if an availability constraint is imposed in one of the first stage's machines, then the following result is verified.

Lemma 1

It is possible that there does not exist any permutation solution to be optimal for $Am|nr - a(M_h)|C_{\max}$.

Proof:

Consider an instance with: $m=2$, $n=2$ and $p_{11} = 4$, $p_{12} = 11$, $p_{21} = 5$, $p_{22} = 1$ and $p_{1A} = p_{2A} = 2$. We have also $s_1 = 4$, $t_1 = 5$.

It can be checked that the unique optimal solution is to sequence J_1J_2 on M_1 , and J_2J_1 on M_2 and M_A . With makespan equal to 14. \square

Although we cannot restrict the research to the permutation solutions, the following result gives us a considerable space reduction.

Lemma 2

For $Am|nr - a(M_h)|C_{\max}$, $1 \leq h \leq m$. There exists an optimal solution where there is the same sequence of jobs on the machines M_k , $k \neq h$ and M_A .

Proof:

Suppose that there is an optimal solution π where the sequence of jobs on a machine M_k for $k \neq h$ is different of the sequence on M_A . Then there exists two jobs J_i and J_j such that J_j is immediately sequenced before J_i on M_k . But J_j follows J_i on M_A .

It can be checked that the inversion of the order of J_i and J_j on M_k will not lead to a higher makespan. And so by applying the same inversion to all the jobs that appears in the wrong order, we can reorder all jobs to satisfy condition of lemma 2. \square

Remark 1: Lemma 2 allows us to reduce the space size from $(n!)^{m+1}$ to $(n!)^2$. From here we will consider only this family of schedules. For notational convenience, and for a given schedule π , the sequence of jobs on the machines M_k , $k \neq h$ and M_A will be denoted simply by π , and the sequence of jobs on the machine M_h will be denoted by π^h .

We present now the worst case analysis.

3.2. Arbitrary heuristic worst case performance

The following theorem gives the bound for a heuristic H giving an arbitrary solution. We note that in such a solution no machine can be idle while a job is ready to be processed on that machine.

Theorem 1

For any heuristic H producing an arbitrary schedule π for $Am|nr - a(M_h)|C_{\max}$, we have

$$C_{\max}(\pi) \leq 3C_{\max}^* .$$

Proof:

Two configurations are to consider:

Case 1: the gap is not part of the critical path, then there exist $1 \leq j \leq n$ and $1 \leq k \leq m$ such that

$$C_{\max}(\pi) = \sum_{i=1}^j p_{\pi_i k} + \sum_{i=j}^n p_{\pi_i A} \leq \sum_{1 \leq i \leq n} p_{ik} + \sum_{1 \leq i \leq n} p_{iA} \leq 2C_{\max}^*$$

Case 2: the gap is a part of the critical path, then there exists $1 \leq j \leq n$ and $1 \leq j' \leq n$ such that:

$$C_{\max}(\pi) = \sum_{i=1}^j p_{\pi_i^h} + t_h - s_h + \Delta + \sum_{i=j'}^n p_{\pi_i A} \leq \sum_{1 \leq i \leq n} p_{ih} + t_h - s_h + \Delta + \sum_{1 \leq i \leq n} p_{iA}$$

Let's note $J_{i_0} = J_h^\pi$. From (1) and (2) we obtain $C_{\max}(\pi_H) \leq 2C_{\max}^* + p_{i_0 h} \leq 3C_{\max}^* . \square$

We now present two heuristics that guaranties a bound of 2.

3.3. H_0 heuristic worst case performance

In the heuristic H_0 we sort the set of jobs according to the processing times on the machine M_h in non-decreasing order. We then consider the resultant permutation solution.

Theorem 2

If we apply the heuristic H_0 to sequence jobs for $Am|nr - a(M_h)|C_{\max}$ then $C_{\max}(\pi) \leq 2C_{\max}^*$ and the bound is tight, where π is the permutation given by the heuristic H_0 .

Proof:

If the gap is not a part of the critical path then, and as explained in the proof of the previous theorem, we have $C_{\max}(\pi) \leq 2C_{\max}^*$.

Otherwise, if the gap is a part of the critical path, then there exists $1 \leq j \leq n$ such that:

$$C_{\max}(\pi) = \sum_{i=1}^j p_{\pi_i h} + t_h - s_h + \Delta + \sum_{i=j}^n p_{\pi_i A} \quad (3)$$

Case 1: if $j = n$ then (3) gives

$$C_{\max}(\pi) = \sum_{1 \leq i \leq n} p_{i h} + t_h - s_h + \Delta + p_{\pi_n A} \quad (4)$$

Let's note $J_i = J_h^\pi$, we know from (2) that $p_{i h} > \Delta$, as we have applied the heuristic H_0 then $p_{\pi_n h} \geq p_{i h} > \Delta$. As $p_{\pi_n h} + p_{\pi_n A} \leq C_{\max}^*$ then we conclude the following from (1) and (4)

$$C_{\max}(\pi) = \sum_{1 \leq i \leq n} p_{i h} + t_h - s_h + \Delta + p_{\pi_n A} \leq \left(\sum_{1 \leq i \leq n} p_{i h} + t_h - s_h \right) + (p_{\pi_n h} + p_{\pi_n A}) \leq 2C_{\max}^*$$

Case 2: If $j \neq n$ then we have $p_{\pi_n h} \geq p_{i h} > \Delta$ and from (3) we get

$$C_{\max}(\pi) = \sum_{i=1}^j p_{\pi_i h} + t_h - s_h + \Delta + \sum_{i=j}^n p_{\pi_i A} \leq \sum_{i=1}^j p_{\pi_i h} + p_{\pi_n h} + t_h - s_h + \sum_{i=j}^n p_{\pi_i A}$$

Then

$$C_{\max}(\pi) \leq \sum_{1 \leq i \leq n} p_{i h} + t_h - s_h + \sum_{1 \leq i \leq n} p_{i A} \leq 2C_{\max}^*$$

In conclusion $C_{\max}(\pi) \leq 2C_{\max}^*$.

The following example shows that the bound is tight. Consider an instance with $m=1$ (i.e. a two machine flow shop problem), $n=2$, $p_{11} = 1$, $p_{21} = z$, $p_{1A} = 2$ and $p_{2A} = 2$. we also have $s_1=z$ and $t_1=z+1$, where z is a positive integer.

If we apply H_0 to the problem we get a sequence J_1, J_2 with $C_{\max}(\pi) = 2z + 3$. However, the optimal solution is to sequence J_2 followed by J_1 with $C_{\max}^* = z + 4$. Hence $C_{\max}(\pi)/C_{\max}^*$ approaches 2 as z approaches infinity. \square

3.4. H_1 heuristic worst case performance

In the heuristic H_1 we apply Johnson's algorithm (JA) to sequence the n jobs by considering only the processing times on the machines M_h and M_A . The optimal permutation obtained for the two machines flow shop problem is then applied for the $Am|nr - a(M_h)|C_{\max}$ problem.

Theorem 3

If we apply the heuristic H_1 to sequence jobs for $Am|nr - a(M_h)|C_{\max}$ then $C_{\max}(\pi) \leq 2C_{\max}^*$ and the bound is tight, where π is the permutation given by the heuristic H_1 .

Proof:

If the gap is not a part of the critical path then, and as explained previously we have $C_{\max}(\pi) \leq 2C_{\max}^*$.

Otherwise, if the gap is a part of the critical path, then there exists $1 \leq j \leq n$ such that:

$$C_{\max}(\pi) = \sum_{i=1}^j p_{\pi_i h} + t_h - s_h + \Delta + \sum_{i=j}^n p_{\pi_i A} \quad (5)$$

Case 1: if $j = n$ then (5) gives

$$C_{\max}(\pi) = \sum_{1 \leq i \leq n} p_{ih} + t_h - s_h + \Delta + p_{\pi_n A} \quad (6)$$

Let's note $J_i = J_h^\pi$, we know from (2) that $p_{ih} > \Delta$. As π_n follows J_i in the sequence then:

$$\min(p_{ih}, p_{\pi_n A}) \leq \min(p_{iA}, p_{\pi_n h}).$$

- If $\min(p_{ih}, p_{\pi_n A}) = p_{ih}$ then $p_{\pi_n h} \geq p_{ih} > \Delta$ so

$$C_{\max}(\pi) = \sum_{1 \leq i \leq n} p_{ih} + t_h - s_h + \Delta + p_{\pi_n A} \leq \left(\sum_{1 \leq i \leq n} p_{ih} + t_h - s_h \right) + (p_{\pi_n h} + p_{\pi_n A}) \leq 2C_{\max}^*$$

- If $\min(p_{ih}, p_{\pi_n A}) = p_{\pi_n A}$ then $p_{\pi_n A} \leq p_{iA}$ so

$$C_{\max}(\pi) = \sum_{1 \leq i \leq n} p_{ih} + t_h - s_h + \Delta + p_{\pi_n A} \leq \left(\sum_{1 \leq i \leq n} p_{ih} + t_h - s_h \right) + (p_{ih} + p_{iA}) \leq 2C_{\max}^*$$

Case 2: if $j \neq n$ then two sub-cases are to consider

Case 2-1: there exists a job $i_2 = \pi_j$ for $j' > j$ such that $p_{i_2 h} \geq p_{ih}$ then $p_{i_2 h} \geq \Delta$. From (5) we get:

$$C_{\max}(\pi) = \sum_{i=1}^j p_{\pi_i h} + t_h - s_h + \Delta + \sum_{i=j}^n p_{\pi_i A} \leq \sum_{i=1}^j p_{\pi_i h} + p_{i_2 h} + t_h - s_h + \sum_{i=j}^n p_{\pi_i A}$$

Then

$$C_{\max}(\pi) \leq \sum_{1 \leq i \leq n} p_{ih} + t_h - s_h + \sum_{1 \leq i \leq n} p_{iA} \leq 2C_{\max}^*$$

Case 2-2: for any job $i_2 = \pi_j$, such that $j' > j$ we have $p_{i_2 h} < p_{ih}$. As J_{i_2} follows J_i in the sequence then: $\min(p_{ih}, p_{i_2 A}) \leq \min(p_{iA}, p_{i_2 h})$. So necessarily $\min(p_{ih}, p_{i_2 A}) = p_{i_2 A}$ and subsequently $p_{i_2 A} \leq p_{i_2 h}$. We can then write

$$\sum_{i=j}^n p_{\pi_i A} \leq p_{\pi_j A} + \sum_{i=j+1}^n p_{\pi_i A} \leq p_{\pi_j A} + \sum_{i=j+1}^n p_{\pi_i h} \quad (7)$$

From (5) and (7) we get

$$C_{\max}(\pi) = \sum_{i=1}^j p_{\pi_i h} + t_h - s_h + \Delta + \sum_{i=j}^n p_{\pi_i A} \leq \sum_{i=1}^n p_{ih} + t_h - s_h + \Delta + p_{\pi_j A}$$

On an other hand, we have: $\min(p_{ih}, p_{\pi_j A}) \leq \min(p_{iA}, p_{\pi_j h})$

- If $\min(p_{ih}, p_{\pi_j A}) = p_{ih}$ then $p_{\pi_j h} \geq p_{ih} > \Delta$ so

$$C_{\max}(\pi) \leq \sum_{1 \leq i \leq n} p_{ih} + t_h - s_h + \Delta + p_{\pi_j A} \leq \left(\sum_{1 \leq i \leq n} p_{ih} + t_h - s_h \right) + (p_{\pi_j h} + p_{\pi_j A}) \leq 2C_{\max}^*$$

- If $\min(p_{ih}, p_{\pi_j A}) = p_{\pi_j A}$ then $p_{\pi_j A} \leq p_{iA}$ so

$$C_{\max}(\pi) \leq \sum_{1 \leq i \leq n} p_{ih} + t_h - s_h + \Delta + p_{\pi_j A} \leq \left(\sum_{1 \leq i \leq n} p_{ih} + t_h - s_h \right) + (p_{ih} + p_{iA}) \leq 2C_{\max}^*$$

In conclusion $C_{\max}(\pi) \leq 2C_{\max}^*$.

The following example shows that the bound is tight. Consider an instance with $m=1$, $n=2$, $p_{11} = 1$, $p_{21} = z$, $p_{1A} = 2$ and $p_{2A} = 2$. We also have $s_1=z$ and $t_1=z+1$, where z is a positive integer. If we apply H_1 to the problem we get a sequence J_1, J_2 with $C_{\max}(\pi) = 2z + 3$. However, the optimal solution is to sequence J_2 followed by J_1 with $C_{\max}^* = z + 4$. Hence $C_{\max}(\pi) / C_{\max}^*$ approaches 2 as z approaches infinity. \square

4. Availability constraint on machine M_A : $Am|nr - a(M_A)|C_{\max}$

In this problem, there is only one availability constraint on the machine M_A . The following result holds:

Lemma 3

For the $Am|nr - a(M_A)|C_{\max}$ problem, there exists an optimal solution which is a permutation schedule.

Proof:

Similar to the proof of lemma 2. \square

Although lemma 3 provides a considerable space reduction, in this case the ratio $C_{\max}(\pi_H)/C_{\max}^*$, where π_H is a solution given by an arbitrary heuristic H , can be arbitrarily large. The following heuristic gives an example:

We consider the heuristic H_{Potts} proposed by [20] for the $Am||C_{\max}$ problem. This heuristic has the tight worst-case ratio bound of $2 - 1/m$. In this heuristic we first apply JA to the $F2||C_{\max}$ problem with the processing time of $J_i, i=1,2,\dots,n$, equal to $(p_{i1} + p_{i2} + \dots + p_{im})/m$ (in the first stage) and to p_{iA} (in the second stage). The found permutation is then applied to the $Am||C_{\max}$ problem.

Lemma 5

If we apply H_{Potts} to sequence jobs for $Am|nr - a(M_A)|C_{\max}$ ($m \geq 2$) then $C_{\max}(\pi_{Potts})/C_{\max}^*$ can be arbitrary large, where π_{Potts} is the permutation given by the heuristic H_{Potts} .

Proof:

Consider an instance with $m=2, n=2, p_{11} = p_{21} = 10, p_{12} = 2, p_{22} = 4, p_{1A} = 7$ and $p_{2A} = 8$. We also have $s_A = 27$ and $t_A = z + 27$, where z is a positive integer.

If we apply H_{Potts} to the problem we get a sequence J_1, J_2 with $C_{\max}(\pi_{Potts}) = z + 35$. However, the optimal solution is to sequence J_2 followed by J_1 with $C_{\max}^* = 27$. Hence $C_{\max}(\pi_{Potts})/C_{\max}^* = (z + 35)/27$ which can be arbitrarily large when z is very large. \square

Remark 2: for the case $m=1$ (i.e. the $F2|nr - a(M_2)|C_{\max}$ problem), H_{Potts} is equivalent to JA , Lee [14] showed that $C_{\max}(\pi_{JA}) \leq 2C_{\max}^*$ and that the bound is tight.

Remark 3: Suppose that the optimal makespan of $Am|nr - a(M_A)|C_{\max}$ ($m \geq 2$) is equal to s_A , and $t_A = s_A + z$ where z is a very large number. Let H be a polynomial heuristic to the problem, unless H gives an optimal solution, we will have $C_{\max}(\pi_H) > t_A > z$, and hence $C_{\max}(\pi_H)/C_{\max}^* > z/s_A$ which can be arbitrarily large. Hence it is impossible to find a polynomial heuristic algorithm with finite error bound (unless $NP = P$).

5. Conclusion

In this paper we have studied the nonresumable model of the two-stage assembly flow shop problem with availability constraint. We have provided two heuristics with tight worst-case ratio bound of 2 when the availability constraint is imposed on one of the first stage's machines. For the case where the gap is imposed on the assembly machine we have shown that no polynomial heuristic algorithm with finite error bound can be found. Future researches include extending the study to the semiresumable model and other performance measures.

References

- [1] I. Adiri, E. Frosting, A.H.G. Rinnooy Kan (1991), Scheduling on a single machine with a single breakdown to minimize stochastically the number of tardy jobs, *Naval Research Logistics*, **38**, 261 – 271.
- [2] R. Aggoune (2004), Minimizing the makespan for the flow shop scheduling problem with availability constraints, *European journal of Processing research* **153**, 534 – 543.
- [3] H. Allaoui, A. Artiba (2006), Scheduling two-stage hybrid flow shop with availability constraints, *Computer and operations research* **33**, 1399 – 1419.
- [4] J. Blazewicz, J. Breit, P. Formanowicz, W.L. Kubiak, G. Schmidt (2001), Heuristic algorithms for the two-machine flow shop with limited machine availability, *Omega* **29**, 599 – 608.
- [5] J. Breit (2004), An improved approximation algorithm for two-machine flow shop scheduling with an availability constraint, *Information Processing letters* **90**, 273 – 278.
- [6] W.J. Chen (2006), Minimizing total flow time in the single-machine scheduling problem with periodic maintenance, *Journal of the Operational Research Society* **57**, 410 – 415.
- [7] T.C.E. Cheng, G. Wang (2000), An improved heuristic for two-machine flowshop scheduling with an availability constraint, *Operation Research Letters* **26**, 223 – 9.
- [8] J.N.D. Gupta (1988), Two-stage hybrid flowshop scheduling problem, *Journal of the Operational Research Society* **39**, 359 – 364.
- [9] M. Haouari, T. Daouas (1999), Optimal scheduling of the 3-machine assembly-type flow shop, *RAIRO Operations Research* **33**, 439 – 445.
- [10] A.M.A. Hariri, C.N. Potts (1997), A branch and bound algorithm for the two-stage assembly scheduling problem, *European Journal of Operational Research* **103**, 547 – 556.
- [11] S.M. Johnson (1954), Optimal two- and three-stage production schedules with setup times included, *Nav. Res. Log. Q.* **1**, 61 – 68.
- [12] W. Kubiak, J. Blazewicz, P. Formanowicz, J. Breit, G. Schmidt (2002), Two-machine flow shop with limited machine availability, *European journal of operational research* **136**, 528 – 540.
- [13] C.Y. Lee (1991), Parallel machines scheduling with non-simultaneous machine available time, *Discrete Applied Mathematics* **30**, 53 – 61.
- [14] C.Y. Lee (1997), Minimizing the makespan in the two-machine flow shop scheduling problem with an availability constraint, *Operations Research Letters* **20**, 129 – 139.
- [15] C.Y. Lee (1999), Two-machine flowshop scheduling with availability constraints, *European journal of operation research* **114**, 420 – 429.
- [16] C.Y. Lee, T.C.E. Cheng, B.M.T. Lin (1993), Minimizing the makespan in the 3-machine assembly-type flowshop scheduling problem, *Management Science* **39**, 616 – 625.
- [17] C.Y. Lee, S.D. Liman (1992), Single machine flow-time scheduling with scheduled maintenance, *Acta Informatica* **29**, 375 – 382.
- [18] C.J. Liao, W.J. Chen (2003), Single-machine scheduling with periodic maintenance and non-resumable jobs, *Computers & Operations Research* **30**, 1335 – 1347.
- [19] C.J. Liao, D.L. Shyur, C.H. Lin (2005), Makespan minimization for two parallel machines with an availability constraint, *European journal of processing Research* **160**, 445 – 456.
- [20] C.N. Potts, S.V. Sevast'janov, V.A. Strusevich, L.N. Van wassenhove, C.M. Zwaneveld (1995), The two-stage assembly scheduling problem: complexity and approximation, *Operations Research* **43**, 346 – 355.
- [21] C. Saffi, B. Penz, C. Rapine, J. Blazewicz, P. Formanowicz (2005), An improved approximation algorithm for the single machine total completion time scheduling problem with availability constraints, *European Journal of Operational Research* **161**, 3 – 10.
- [22] S. Xi, M. Kazuko, N. Hiroyuki (2003), Powerful heuristics to minimize makespan in fixed, 3-machine, assembly-type flowshop scheduling, *European Journal of Operational Research* **146**, 498 – 516.