An Improved Solution Algorithm for Two-Job Shop Scheduling Problems with Availability Constraints

Riad Aggoune [*]

Abstract—**This paper presents a reduced-complexity algorithm for exactly solving two-job shop scheduling problems where an arbitrary number of non-availability periods may occur on each of the m machines ($m > 2$). Considering the limited availability of machines makes the scheduling models more realistic in comparison to usual ones. The proposed solution algorithm is an extension of the geometric approach developed for the classical two-job shop problem. It is based on a new representation that allows dealing with modified processing times and thus considering unavailability constraints of several kinds.**

Keywords: Shop scheduling, Availability constraints, Geometric approach, Complexity

1 Introduction

Integrating practical constraints into traditional models is an important issue for researchers and practitioners in scheduling. Among those realistic considerations is the limited availability of the machines. Indeed in actual workshops planning horizons may contain time periods corresponding to preventive maintenance tasks or dedicated to the execution of urgent orders. Those unavailability constraints may prevent or postpon the ordinary execution of jobs, and must be taken into account when building schedules.

During the last decade a fast growing number of studies were dedicated to scheduling problems with machine availability consideration. We may refer to the state of the art proposed in [12]. Two-machine flow shop problems were studied in [4] and [7], who established complexity and performance guarantee results. In [10], several one machine and the associated job shop problems were addressed. Several kinds of availability constraints are defined in the scheduling literature. In the strictly non-preemptive model [1], the execution of an operation can be interrupted neither by an unavailability period nor by another operation. In [8] and [9] the author defined three models of availability constraints, namely the resumable, non-resumable and semi-resumable cases. In the latter, it is possible to execute a part of an operation before the unavailability of a machine (hole) and then to complete its execution after the hole, possibly with a penalty proportional to the already executed part. When there is no penalty, operation are said to be resumable, whereas a maximal penalty corresponds to the non-resumable model. In [10], the authors defined by crossable availability periods the ones that allow jobs to be preempted. Flexibility on the availability constraints was introduced in [1]. In that study the maintenance activity is not fixed in advance but must be determined during the scheduling procedure. Time windows in which their position may vary are thus assigned to maintenance tasks. More recently, authors in [11] proposed heuristics for the permutation flow shop problem, where the machines are not available at the beginning of the planning period.

In this paper, we present a new polynomial algorithm for scheduling job shops with two jobs and availability constraints. More precisely, we assume that there can be an arbitrary number of non-availability periods on each of the m machines (with $m > 2$). The unavailability periods, due to preventive maintenance activity for instance, are supposed to be known in advance and fixed. The goal of the algorithm is to schedule operations of the two jobs in order to minimize the schedule length. The remainder of this paper is organized as follows. We first present the considered problems in Section 2. Section 3 describes in details the temporized geometric approach that allows exactly scheduling two-jobs with non-preemptive availability constraints. The new solution algorithm and its direct extensions to general availability models are finally presented in Section 4. Section 5 concludes the paper.

2 Problem description

The job shop scheduling problem with two jobs and availability constraints can be stated as follows: Two jobs J_1 and J_2 must be executed on a set of m machines $M = \{M_1, M_2, ...M_m\}$. Each job J_i consists of a linear sequence of n_i operations $M = \{O_{i,1}, O_{i,2}, ...O_{i,n_i}\}$. Each machine can perform at most one operation at a

[*]Public Research Centre Henri Tudor, Business & Systems Analytics Unit, 6, rue de Luxembourg, BP144, L-4002 Esch-sur-Alzette Tel/Fax: +352.42.59.91.545/555 Email: riad.aggoune@tudor.lu

time and each operation $O_{i,j}$ of job J_i needs one machine M_j at a time, during $p_{i,j}$ time-units. An arbitrary number K_j of unavailability periods may occur on each machine M_j. Let K be the maximal number of holes, i.-e. $K = max_{j=1,...,m}\{K_j\}$. Starting time and duration of these tasks are known in advance and fixed. Objective is to determine the execution dates for the operations such that to above constraints are satisfied and the makespan is minimized. According to the notation introduced in [12], the scheduling problem can be denoted $J, NC_{win}\,|n=2|\,C_{max}$, where NC_{win} means that unavailability periods are arbitrarily distributed on the machines. It is shown in [2], that the strictly non-preemptive scheduling problem is polynomial and its complexity is equal to $O(Ks^4)$, where $s = max\{n_1, n_2\}$. The solution algorithm applies for any regular criteria. An extension of this work to the resumable case is also proposed. In what follows we show that the complexity of problem $J, NC_{win}\,|n=2|\,C_{max}$ can be improved and that the new polynomial algorithm applies not only to the resumable model but to a general model including all cases of fixed availability constraints. This general model was introduced in [3].

We associate with each operation $O_{i,j}$ a coefficient $\alpha_{i,j}$ which represents the proportion of operation $O_{i,j}$ to redo after the unavailability period interrupting the operation. It represents its resumable character. Thus $\alpha_{i,j} = 0$ if $O_{i,j}$ is resumable, $\alpha_{i,j} = 1$ if $O_{i,j}$ is non-resumable and $0 \leq \alpha_{i,j} \leq 1$ if $O_{i,j}$ is semi-resumable. We also associate coefficient $\beta_{i,j,k}$ to an operation $O_{i,j}$ which must be processed on machine M_r and which can be interrupted by an unavailability period $h_{r,k}$. It represents the preemptive character of $O_{i,j}$ or the crossable character of availability period $h_{r,k}$. Thus, $\beta_{i,j,k} = 0$ if $h_{r,k}$ is non-crossable or $O_{i,j}$ is strictly non-preemptive. In this case, there is a disjunction between $O_{i,j}$ and $h_{r,k}$. On the opposite, $\beta_{i,j,k} = 1$ if $h_{r,k}$ is crossable and $O_{i,j}$ is preemptive. In this case, the position of $O_{i,j}$ in relation to $h_{r,k}$, depends on the resumable character of $O_{i,j}$.

3 Temporized geometric approach

In this section, we present the temporized geometric approach proposed in [2], which allows solving two-job shop scheduling problems with availability constraints. For sake of clarity, we first describe the classical geometric approach for the makespan minimization and give complexity results. Then, we explain how the algorithm is modified to integrate non-premptive availability constraints. For sake of clarity, we use example the following example in the remainder of the paper to illustrate the approaches. There are two jobs to be scheduled on four machines. The manufacturing processes of jobs are as follows:
$J_1 = \{M_1(2), M_2(4), M_3(2), M_4(1)\}$ and $J_2 = \{M_3(1), M_1(2), M_2(3), M_4(2)\}$.

3.1 The geometric approach

The idea behind the graphical approach consists in reducing the two-job shop scheduling problem in a shortest path one (see for instance [5] and the references therein). The approach starts by representing the scheduling problem in a two-dimension plane with rectangular objects as obstacles defined as follows:

- Each axis corresponding to one job is decomposed in n_i sub-intervals according to the manufacturing process of the job. A sub-interval $I_{i,j}$ corresponds to operation $O_{i,j}$ (i.e. operation j of job J_i) and its length is equal to the processing time of that operation.
- A rectangle defined by sub-intervals I_{1,k_1} and I_{2,k_2} is an obstacle, denoted as (k_1, k_2), if and only if operations O_{1,k_1} and O_{2,k_2} use the same machine.
- The upper and right boundaries of the rectangle defined by the origin O (which indicates the start of processing of the two jobs) and the final point $F = (\sum_{k=1}^{n_1} p_{1,k}, \sum_{k=1}^{n_2} p_{2,k})$ (which points out the finish of processing of the two jobs) is considered as a degenerated obstacle (see Figure 1).

In Figure 1, the rectangle defined by $I_{1,1}$ and $I_{2,2}$ is an obstacle as the first operation of J_1 and the second operation of J_2 are both performed on machine M_1.

A feasible solution of the two-job shop scheduling problem corresponds to a path going from the origin O to the final point F, which avoids the interior of any obstacle (k_1, k_2) seeing that operations O_{1,k_1} and O_{2,k_2} are performed on the same machine and the preemption is not allowed. The path is composed of horizontal segments where only job J_1 is processed (segment HS in Figure 1), vertical segments where only job J_2 is processed (segment VS in Figure 1) and diagonal segments where jobs J_1 and J_2 are performed concurrently using different machines (segment DS in Figure 1). The length L of a path, which is equal to the duration of the associated schedule, is $L = \sum(\text{horizontal segments}) + \sum(\text{vertical segments}) + (1/\sqrt{2})\sum(\text{diagonal segments})$.

Accordingly, the determination of the optimal schedule for the $J\,|\,n=2\,|\,C_{max}$ becomes a shortest path problem in the two-dimension plane, which consists of finding a shortest trajectory that connects the origin O to the final point F. This shortest path problem can equivalently be obtained in a digraph $G = (V, E, d)$ defined as follows:

- The set of nodes V is composed, in addition to the origin O and the final point F, of South-East (SE) and North-West (NW) corners of obstacles met when progressing diagonally in the plane.

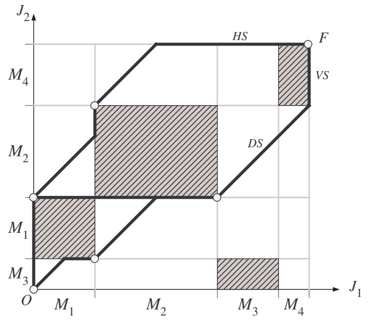

Figure 1: The shortest path in the plane

- Each node $v_i \in V - \{F\}$ has at most two outgoing arcs. These arcs are obtained by progressing diagonally (45°) until an obstacle (k_1, k_2) is hit. If the obstacle (k_1, k_2) is the final obstacle, the node F is the only successor of v_i, otherwise the SE and NW corners of (k_1, k_2) are immediate successors of v_i (see Figure 1). Hereafter, the south-east corner of an obstacle (k_1, k_2) is denoted $SE_{(k_1,k_2)}$ and the north-west corner is denoted $NW_{(k_1,k_2)}$.
- The distance between v_i and a south-east (resp. north-west) corner is the projection along the x (resp. y) axis.

In Figure 1, diagonal progress from $NW_{(2,3)}$ hits the upper boundary of the final obstacle and hence the only successor of $NW_{(2,3)}$ is the node F. Diagonal progress from $SE_{(1,2)}$ hits the obstacle $(2,3)$ and thus the $SE_{(2,3)}$ and $NW_{(2,3)}$ are its two immediate successors. The path $((O, SE_{(1,2)}), (SE_{(1,2)}, SE_{(2,3)}), (SE_{(2,3)}, F))$ is the shortest path of length 11 corresponding to perform J_1 and J_2 in parallel until instant $t = 1$, finish $O_{1,1}$ while J_2 waits, perform J_1 and J_2 in parallel between $t = 2$ and $t = 4$, finish $O_{1,2}$ while J_2 waits, perform again J_1 and J_2 in parallel between $t = 6$ and $t = 9$, and then finish with the last operation of J_2.

The graphical approach can be extended to solve the two-job shop scheduling problem with any regular criterion $\Phi(C_1, C_2)$ where $C_i (i = 1, 2)$ is the completion time of job J_i. This extension has been proposed in [14] and consists in evaluating the criterion each time the upper or right boundary of the final obstacle is reached, i.-e when a job is completed.

The complexity of the graphical approach is based on the property that the constructed digraph is sufficient to find an optimal solution of $J \mid n = 2 \mid \Phi$ ([5]) and can be ontained in $O(r \log r)$ steps where r is the number of obstacles ([13], and [5]). The graphical approach can be extended to solve the two-job shop scheduling problem with additional constraints, namely when preemption [14], precedence constraints and release dates [6] are considered.

3.2 Temporized approach

The temporized geometric approach (TGA) proposed in [2] is a polynomial algorithm for solving the two-job shop scheduling problem with availability constraints in the strictly non-premptive case. TGA is mainly based on the definition of earliest dates associated with each vertex of the network representing the problem. This characterization allows to integrate the evolution of time and thus to deal with the limited availability of the machines. More precisely, we compute for each vertex v of the plane with obstacles the earliest date, noted $h(v)$, and which corresponds to the smallest duration to reach vertex v from the origin O (note that $h(O) = 0$). Thus, $h(v)$ is equal to the length of the shortest path going from O to v. Tests are realized each time an operation has to be started, i.-e. each time a path crosses a vertical or horizontal line, so as to verify if the machine necessary for its execution is available. Knowing the earliest date $h(v)$ of vertex v, an availability test on direction J_1 (resp. J_2) consists in determining the availability time $T_{1,h(v)}$ (resp. $T_{2,h(v)}$) of the machine associated with the next operation of job J_1 (resp. J_2).

In Figure 2, operation $O_{1,j}$ job J_1 cannot be processed at time $h(v) = 2$, since it cannot be completed before the first unavailability period of the machine. Operations being strictly non-premptable, the execution of $O_{1,j}$ can start (in the earliest) at time $T_{1,h(v)} = 8$ which is equal to the ending time of the first unavailability period.

Starting from a given vertex v with earliest date $h(v)$, several ways to progress and determine its successors are to be considered:

- If the operations of the two jobs cannot start at time $h(v)$, vertex v is duplicated. A waiting vertex v_w is created and v_w is the only successor of v.
- If there is an availability problem in direction J_1 (resp. J_2), the progression is made along the vertical (resp. horizontal) line, that corresponds to only

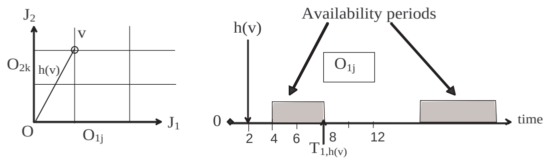

Figure 2: Availability test on direction J_1

execute the operations of job J_2 (resp. J_1), until job J_1 (resp. J_2) becomes available. A singular vertex s, from which a diagonal progression is possible, is added as the unique successor of v.

- If there is no availability problem, i. -e. if the operations of the two jobs can be executed at time $h(v)$, the progression works as in the classical geometric approach: It is made in the diagonal direction until an obstacle is hit. If this obstacle represents a machines conflict, its NW and SE corners are the two direct successors of vertex v. If the hit obstacle is the final one, point F is added as the unique successor of v.

In [2], it is proved that network N composed by the points listed above is sufficient to find an optimal solution for problem $J, NC_{win}\,|n=2|\,C_{max}$, and that N can be obtained in $O(Ks^4)$ steps, where s is the number of obstacles, equal to $O(n_1.n_2)$. Then, the complexity of problem $J, NC_{win}\,|n=2|\,C_{max}$ is $O(Ks^4)$. Using the same proceed as in [14], it can be shown that the results applies to any regular criterion $\Phi(C_1, C_2)$. The resumable problem is also sovable in $O(Ks^4)$ steps [2]. However, the temporized geometric approach does not apply to the semi-resumable case. Finally, the consideration of head and tails, and precedence constraints as in [6] is also possible and problem $J, NC_{win}\,|\,n=2\,;\,prec\,;\,r_i\,|\,\max(C_{i,j}+q_{i,j})$ is solvable in $O(Ks^6)$ steps, what provides lower bounds for the job shop scheduling problem with availability constraints [2]. Note that the increasing in the complexity of TGA regarding the classical geometric approch is due to the introduction of new points, namely singular and waiting vertices, during the construction of the network associated with the search of a shortest path.

4 Improved approach

We develop in this section the improved approach for problem $J, NC_{win}\,|n=2|\,C_{max}$ including all cases of fixed availability constraints. As the TGA algorithm, it is based on the introduction of earliest dates for the vertices of the network associated with the problem. We propose a new definition of availability tests that allows to integrate the actual availability of the machines without introducing additionnal points. Indeed, we propose not to focus on the time an operation can start according to the availability of the involved machine but the time needed to complete the operation. It is thus necessary to modify the way the graphical representation is defined.

4.1 New Representation

In the proposed approach we keep the classical representation in the plane with obstacles of the scheduling problem. However, the two axis are decomposed into unitary sub-intervals $I_{i,j}$ which lenght are no more proportional to the processing times of the associated operations $O_{i,j}$. Instead, we propose to divide the sub-intervals according to the processing times. As a consequence, the simultaneous execution of two operations $O_{1,j}$ and $O_{2,k}$ job J_1 and J_2 cannot be not represented by a diagonal leg (with an angle of 45 deg), but follows a direction induced by the processing times of the involved operations. Let us consider the following example, illustrated in Figure 3, where one operation $O_{1,j}$ of job J_1 and two operations $O_{2,k}$ and $O_{2,k+1}$ of jobs J_2 are represented by unitary intervals. The processing times are respectively equal to $p_{1,j}=4$, $p_{2,k}=2$ and $p_{2,k+1}=5$. Starting from vertex v, the simultaneous execution of $O_{1,j}$ and $O_{2,k}$ corresponds to a progression in the direction given by angle α.

Since $p_{1,j} > p_{2,k}$, the execution of $O_{2,k}$ finishes before the one of $O_{1,j}$. Thus, the path crosses the upper boundary of the square corresponding to the operations. When operation $O_{2,k+1}$ can start, $\Delta_{1,j}=2$ units of $O_{1,j}$ are already executed. As in the classical geomtric approach, the minimal quantity between $p_{2,k+1}$ and $p_{1,1}-\Delta_{1,j}=2$ can be executed before the path crosses the next horizontal or vertical line.

Angle α is such that the length of its opposite side $L_{2,k}$,which correponds to the progression of job J_2 is:

$$L_{2,k} = \frac{min(p_{1,j}-\Delta_{1,j}, p_{2,k}-\Delta_{2,k})}{p_{2,k}}, \qquad (1)$$

where $\Delta_{2,k}$ and $\Delta_{1,j}$ represent respectively the portion of operation $O_{1,j}$ and $O_{2,k}$ already executed. Note that these values are both equal to 0 for the classical vertices. However, it is necessary to evaluate the direction of the progression each time the path crosses a horizontal or vertical line. In fact in this case only one operation is to be started, the other one being partially executed.

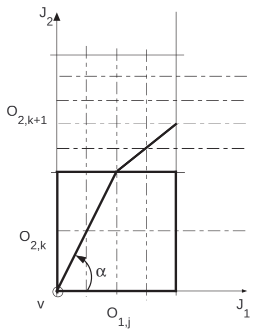

Figure 3: New progression

The lenght $L_{1,j}$ of the adjacent side of angle α, which corresponds to the execution of job J_1 is given by:

$$L_{1,j} = \frac{min(p_{1,j} - \Delta_{1,j}, p_{2,k} - \Delta_{2,k})}{p_{1,j}}. \quad (2)$$

Finally, angle α is defined by:

$$tan\alpha = \frac{L_{2,k}}{L_{1,j}} = \frac{p_{1,j}}{p_{2,k}}. \quad (3)$$

In the example considered in Figure 3, the path progressing from v first goes in direction $\alpha = arctan(\frac{p_{1,j}}{p_{2,k}}) = arctan(\frac{4}{2}) \approx 63deg$. Then, the direction is given by $arctan(\frac{p_{1,j}}{p_{2,k+1}}) = arctan(\frac{4}{5}) \approx 39deg$.

Figure 4 (left) shows the new representation for the problem given by previous example and where no availability constraint is considered. Numbers represent the earliest dates associated with the vertices. We can see that the set of vertices is the same as in Figure 1. Indeed, the new representation respects the rates of executions between operations of the two jobs. Pathes hit the obstacles at the same portions of operations. The main difference with the classical graphical approach is that it is no more possible to read the distances between vertices by (orthogonal) projections of the legs on the axis. However, we are more interested in the time needed to reach a vertex v' starting from vertex v, than in the geometrical distance between them. That information is already included in the definition of the earliest dates of the vertices. The general distance, which is the length of the shortest path between two vertices v and v' is thus defined by formula $\tilde{d}(v, v') = h(v') - h(v)$.

4.2 Availability Constraints

The main advantage from using this new representation is that it allows dealing with modified processing times and thus easily integrate availability constraints in the scheduling problem. In fact, the only additional step is to suppose that the processing times of operations are not fixed in advance but depend on their starting times (which are given by the earliest dates $h(v)$ of the considered vertices) and the availability of the machines. Given a vertex v with earliest date $h(v)$, we compute the time needed to complete next operations of job J_1 and J_2.

The new availability tests we propose thus consist in finding the modified processing times of operations. In the stricly non-preemptive model, the new processing times include the initial ones and possibly an iddle time (corresponding to the time preceding an availability period) and the duration of an availability period. In the example provided by Figure 2, the modified processing time of operation $O_{1,j}$ is $\tilde{p}_{1,j} = 10$. More precisely, $\tilde{p}_{1,j} = (S_{r,1} - h(v)) + P_{r,1} + p_{1,j}$, where $S_{r,1}$ and $P_{r,1}$ are respectively the starting time and the duration of the first unavailability period of the considered machine M_r.

When the modified processing times $\tilde{p}_{1,j}$ and $\tilde{p}_{2,k}$ are calculated for operations $O_{1,j}$ and $O_{2,k}$ of jobs J_1 and J_2 respectively, the progression is done in the direction given by angle α such that:

$$tan\alpha = \frac{\tilde{p}_{1,j}}{\tilde{p}_{2,k}}. \quad (4)$$

For instance, let suppose that machine M_3 is unavailable between times $S_{3,1} = 7$ and $S_{3,1} + P_{3,1} = 9$ in the considered example. The first part of the shortest path given by Figure 4 remains the same. If we consider the last two operations of jobs and vertex v which earliest date is $h(v) = 6$, then processing time of operation $O_{1,3}$ of job J_1 becomes $\tilde{p}_{1,3} = (S_{3,1} - h(v)) + P_{3,1} + p_{1,3} = (7-6) + 2 + 2 = 5$. The other processing times remain unchanged. Thus, the path from vertex v goes first in the direction given by angle $\alpha = arctan(\frac{\tilde{p}_{1,3}}{p_{2,3}}) = arctan(\frac{5}{3})$. Then the path crosses the horizontal line corresponding to the last operation of job J_2. The new direction for the path is given by $\alpha = arctan(\frac{\tilde{p}_{1,3}}{\tilde{p}_{2,4}}) = arctan(\frac{5}{2})$. The next Figure 4 (right) shows the modifications of the shortest path. It is easy to see that to the general model including all cases of fixed unavailability periods can be addressed the same way. Indeed, the modified processing time for

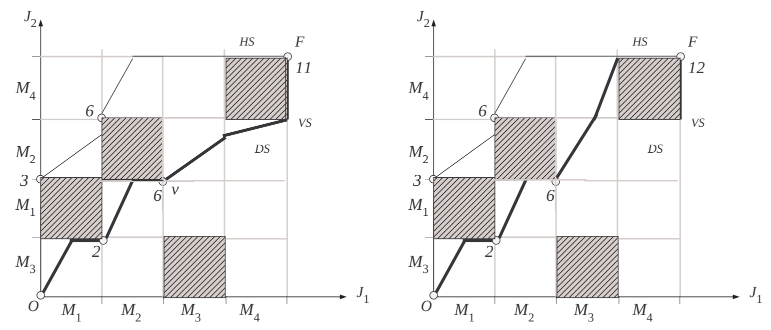

Figure 4: New representation, without (left) and with (right) availability constraints

an operation $O_{i,j}$ which starts at time $h(v)$ and cannot be completed before the starting time $S_{r,k}$ of an unavailability period $h_{r,k}$ is obtained by the general formula:

$$\tilde{p}_{i,j} = (S_{r,k}-h(v))+P_{r,k}+p_{i,j}+\beta_{i,j,k}(\alpha_{i,j}-1)(S_{r,k}-h(v)). \quad (5)$$

$$= P_{r,k} + p_{i,j} + [1 + \beta_{i,j,k}(\alpha_{i,j} - 1)](S_{r,k} - h(v)).$$

5 Conclusions and Future Work

We proposed in this paper a new polynomial algorithm for solving two-job shop scheduling problems with availability constraints. This algorithm outperforms existing one not only in term of complexity but also regarding its applications. Indeed, the new algorithm solves two-jobs problems for all fixed availability constraints models. A problem that remains open is wether a similar approach can be developed in case of flexible availaibility constraints.

References

[1] Aggoune, R., "Minimizing the Makespan for the Flow Shop Scheduling Problem with Availability Constraints," *European Journal of Operational Research*, V153, pp. 534-543, 2004.

[2] R. Aggoune and M.-C. Portmann. "Flow shop scheduling problem with limited machine availability: A heuristic approach," *International Journal of Production Economics*, V99, pp. 4-15, 2006.

[3] Azem, S., Aggoune, R., Dauzere-Peres, and S., "A mathematical model for job shop scheduling with resource availability constraints," *International Conference on Industrial Engineering and System Management (IESM 2007)*, Beijing, China, May 2007.

[4] Blazewicz, J., Breit, J., Formanowicz, P., Kubiak, W., Schmidt, G., "Heuristic algorithms for the two-machine flow-shop problem with limited machine availability," *Omega Journal*, 29, 599–608, 2001.

[5] Brucker, P., "An efficient algorithm for the job-shop problem with two jobs," *Computing*, 40:353–359, 1988.

[6] Brucker, P., Jurisch, B., "A new lower bound for the job-shop scheduling problem," *European Journal of Operational Research*, V64, pp 156-167, 1993.

[7] Kubiak, W., Blazewicz, J., Formanowicz, P., Breit, J., Schmidt, G., "Two-machine flow shops with limited machine availability," *European Journal of Operational Research*, V136, pp. 528-540, 2002.

[8] Lee, C.-Y., "Two-machine flowshop scheduling with availability constraints," *Journal of Global Optimization*, V9, pp. 395-416, 1996.

[9] Lee, C.-Y., "Minimizing the makespan for the two-machine flowshop scheduling with availability constraints," *European Journal of Operational Research*, V114, pp. 420-429, 1999.

[10] Mauguire, P., Billaut, J-C., Bouquard, J.-L., "New single machine and job-shop scheduling problems with availability constraints," *Journal of Scheduling*, V8, pp. 211-231, 2005.

[11] Perez-Gonzalez, P., Framinan, J. M., "Scheduling permutation flowshops with initial availability constraint: Analysis of solutions and constructive heuristics," *Computers and Operations Research*, V36, pp. 2866-2876, 2009.

[12] Schmidt, G., "Scheduling with limited machine availability," *European Journal of Operational Research*, V121, pp. 1-15, 2000.

[13] Sotskov, Y.N., "Optimal servicing two jobs with a regular criterion," *in: Automation of Designing Processes*, Minsk, pp. 86-95, 1985 (in Russian).

[14] Sotskov, Y.N., "The complexity of shop-scheduling problems with two or three jobs," *European Journal of Operational Research*, V53, pp. 326-336, 1991.