Journal of Industrial Engineering International
January 2009, Vol. 5, No. 8, 52-60

Islamic Azad University, South Tehran Branch

Flexible job shop scheduling under availability constraints

Mohammad Taghi Taghavi-Fard*

Assistant Professor, Postgraduate Engineering Center, Islamic Azad University, Tehran, Iran

Hamid Reza Dehnar Saidy

M.Sc., Young Researchers Club, Islamic Azad University, Science and Research Branch, Tehran, Iran

Abstract

In this paper, an exact geometric algorithm is presented for solving two-job sequencing and scheduling problems in flexible flow shop and job shop environments while the resources are (un)available in some time periods and processors (un)availability is the same in all work centers. This study seems utterly new and it is applicable to any performance measure based on the completion time. The investigated models are very close to the actual scheduling problems, because they envisage the flexible job shop environments, *heads*, *set-up times*, arbitrary number of unavailability periods on all resources, *arbitrary number of work-centers*, any kind of cross-ability, any kind of resume-ability and *several types of performance measures*. The proposed model is presented to solve two-job problems because it is a graphical approach. However, it is concluded that the idea can be extended to n-dimensional problems as well.

Keywords: Flexible job shop scheduling; Resumeable operations; Akers' graphical algorithm; Availability constraint

1. Introduction

Scheduling is the process of assigning activities to resources over time. Jobs are *sequenced* based on the problem performance measure. A variety of constraints such as: duration of activities, release and due dates, precedence constraints, and resource availability might affect the scheduling problem.

Most of the scheduling literature reviews are based on the assumption that machines are continuously available. This assumption might be justified in some cases, but in real world, continuous availability of a machine is not usually possible. The machine might not be available due to a deterministic or a random reason. This limited availability of machines might result from preschedules, preventive maintenance, or the overlap of two consecutive time horizons in the rolling time horizon planning algorithm. The rolling horizons are used mainly, because most of the real world problems of production planning are dynamic. On the other hand, the input data are being frequently updated. A time period in which a machine is unavailable has been named a *hole* for convenience (see Kubiak et al. 2002).

In this paper, a graphical algorithm is presented for scheduling with limited resource availability in flexible job shop environments. The investigated model is an offline deterministic and static system with any performance measure based on completion time. We have focused on two-job (event) and arbitrary number of resources sequencing problem. In deterministic models, the processing times, release and due dates of the jobs and the starting time and the duration of the unavailability period are known at time zero. However, pre-emption is allowed in presented algorithm.

Flexible job-shop is an *environment* in which the number of work-centers (*dissimilar* resources) is greater than one, processing order is variable for all the jobs (tasks), job-order isn't the same for all resources and the available number of identical resources (identical processors) is greater than 1 at least in one workcenter. Thus, the *flexible job-shop scheduling* (FJS) problem concerns two sub-problems: i) assignment of each operation to one of the alternative machines (*assignment sub-problem*); ii) ordering of the operations on each assigned machine (*sequencing sub-problem*), with the aim of optimizing an objective function.

* *Corresponding author. E-mail: dr_taghavifard@yahoo.com*

As compared to the classical scheduling issues, studies dealing with limited machine availability are very rare. Availability constraint problem was first introduced for parallel (flexible single) machines [18,19] and single machine [1] environments. Lee extensively investigated flow-shop scheduling problems with two machines [10,11,12]. In particular, he defined the *resumeable*, *non-resumeable* and *semi-resumeable* models. An operation is called resumeable if it can be interrupted by an unavailability period and completed without penalty as soon as the machine becomes available again. In this case, it is said that "pre-emption is allowed". When the processing of an operation cannot be interrupted by an unavailability period and must be done by one-shot, then that operation is called non-resumeable. If a part of an operation - which has been processed before the unavailability period - must be *partially* re-executed, then the operation is called semi-resumeable. The resumeable, non-resumeable and semi-resumeable terms are applied to job and availability constraint problems. Balas et al. single machine algorithm is used in a Shifting Bottleneck Procedure to solve job shop scheduling problems with deadlines [5]. Their algorithm can also be used to provide an approximate solution for job shop non-resumeable problems with the greatest completion time of an operation as the performance measure [7]. Non-resumeable job-shop problem for minimization of the makespan was solved by Aggoune (2002) using a branch and bound (B&B) algorithm. Aggoune (2004) extended Akers' graphical approach to the job shop non-resumeable problems for minimization of the makespan with arbitrary holes on all machines. He named his method as *temporized geometric approach* (*TGA*) that is a polynomial algorithm. Mauguière et al. have proposed a B&B algorithm with arbitrary holes to minimize the makespan for solving job shop resumeable problems [14]. Computational results show that this problem is a little more difficult to solve than the problem without unavailability periods. Aggoune extended his non-resumeable "*TGA*" to a resumeable problem that it is still polynomial and its complexity is the same [3].

Mauguière et al. have proposed a B&B algorithm with arbitrary holes for *crossable* job shop *resumeable / non-resumeable* problems with makespan as the measure of performance [16]. If some operations are resumeable and the others are not, availability constraint is named *resumeable* or *non-resumeable*, respectively. An unavailability period which allows an operation to be interrupted and resumed after a specific time period is called "crossable" while an unavailability period that prevents the interruption of any operation, even if the operation is resumeable, is called "*non-crossable*". When some unavailability periods are crossable and the others are not, the problem is called *crossable* or *non-crossable,* respectively [15]. Mauguière et al. extended their job-shop algorithms to crossable/non-crossable and resumeable/non-resumeable with arbitrary holes on all machines considering ready times and makespan as well [15]. However, there is a very little theoretical study in the scheduling literature for the flexible job shop scheduling problems with arbitrary resources and availability constraints.

We have used a geometric approach for solving the considered problems. Geometric approach was presented by Akers and Friedman in 1955 for the first time [4]. It consisted to minimize the maximum flow time of two jobs in a flow shop environment by shortest path method in a two-dimensional graph. Then, Akers (1956) introduced "Akers' graphical method" for job shop production scheduling problems [17]. As it is stated by Makui, Hardgrave and Newhauser completed this algorithm to minimize makespan [13]. The complexity of this two-job shop scheduling problem for the minimization of any regular criterion is stated by Sotskov and Brucker [6,20]. Later on, Mensch extended Akers' algorithm for "arbitrary" jobs [17].

In Section 2, the characteristics of the machine scheduling problem with availability constraints is defined. The considered algorithm and its extension (by adding set-up times to the model) are presented in Section 3 and Section 4, respectively. In Section 5, other extensions of the debated model are described and in Section 6, we wrap up the subject by presenting our conclusion.

2. Notations

The following notations will be used throughout the paper:

i	Job index; i = 1, 2, …, n (n = number of jobs).
j	Work center index; j = 1, 2, …, m (m = number of work centers).
J_i	Job (event or task) i.
R_j	Resource (processor) j.
p_{ij}	Processing time of J_i on R_j .

r_i Ready time (release date or head) of J_i.

w_i Weight of J_i.

f_i Flow time of J_i.

C_i Completion time of J_i $(C_i = f_i + r_i)$.

K Total number of holes (unavailability periods) in the system.

s_j^k Starting time of k^{th} unavailability period in j^{th} work center.

e_j^k Ending time of k^{th} unavailability period in j^{th} work center.

m_j Number of identical processors in j^{th} work center.

FJ Flexible job shop environment.

h_{Kj} Number of *K* holes on all resources.

rs Resumeable operation in arbitrary unavailability pattern.

nr Non-resumeable operation in arbitrary unavailability pattern.

sr Semi-resumeable operation in arbitrary unavailability pattern.

rs/nr/sr System with resumeable, nonresumeable and semi-resumeable operations.

cr System with crossable holes.

ncr System with non-crossable holes.

cr/ncr System with both crossable and non-crossable holes.

S_{ij} Set-up time of R_j for J_i.

μ_{ij} Fraction of processing time (p_{ij}) before a hole that must be repeated after that hole for semi-resumeable operation $(0 < \mu_{ij} < 1)$.

γ Performance measure.

F Total flow time of all jobs $(F = \sum_{i=1}^{n} f_i)$.

$\overline{C_w}$ Mean weighted total completion time of all jobs $(\overline{C}_w = \sum_{i=1}^{n} w_i C_i / \sum_{i=1}^{n} w_i)$.

$C_{\max}$ Makespan $C_{\max} = \max(C_i; i = 1,...,n))$.

A Operation accomplishment (attainment) time of all resources.

W Total waiting time $(W = F - \sum_{j=1}^{m}\sum_{i=1}^{n} P_{ij})$.

The considered problem is generally *FJ*, h_{Kj}, *cr/ncr*| $n = 2$, *rs/nr/sr*, r_i, $m_j \le 2$ | γ, that denotes a flexible job shop problem with 2 jobs by crossable and non-crossable arbitrary unavailability pattern and any kind of cross-ability having different ready times, arbitrary number of work centers, in which there are 1 or 2 identical processors and *K* holes, and objective function is minimizing γ (γ is *F*, $\overline{C_w}$, $C_{\max}$, *W* or their equivalent criteria). All identical processors may or may not be available in a specific time period.

3. Proposed new graphical algorithm

In this section, we have extended Akers' graphical method which exactly solves the problem of *FJ*, h_{Kj}, *cr*| $n = 2$, *rs*, r_i, $m_j \le 2$ | γ, where γ is completion time. Note that, the starting and ending times are supposed to be the same for both resources if there are two identical processors in a work center. It is clear that for $m_j \ge 2$, jobs sequencing algorithm is not needed because, optimum solution is constant for any sequence.

Our proposed method is described as follows:

Step 1. Draw a two dimensional coordinates. The horizontal and vertical axes show the operation times of J_1 and J_2 in all work centers, respectively.

Step 2. Show each of the process times of any work center by a rectangle. Differentiate among the rectangles related to work centers having 1 processor and the others. The horizontal length of the rectangle R_j is equal to p_{1j}. The vertical width of the rectangle R_j is equal to p_{2j}. The order of rectangles on any axis is the same in the work centers. The southwest corner of any rectangle situates in as early as possible. This time corresponds to 3 items:

- North-east (NE) corner of the prior rectangle in job routing (for the first rectangle it will be coordinates origin),
- r_1 and r_2,
- Starting and ending times of related holes (see Figure 1). In *rs* operation, if p_{ij} is greater than the duration of the first availability period, rectangle R_j cannot be placed in the first availability period. So, break it down into as many rectangles as needed and fit them into the first available time period. In this case, the total length of all rectangles R_j must be equal to p_{1j} and total width of all rectangles R_j must be equal to the p_{2j}. These rectangles are set tandem. But in *nr* operation, rectangle R_j is completely placed in the first availability period that its duration isn't smaller than p_{ij}.

Step 3. Start to move from origin coordinates (*O*) to (*D*). *D* is NE corner made by north side of last rectangle for J_2 and east side of last rectangle for J_1. There are 3 possible moves at any point: horizontal, vertical and diagonal (45° line). Move only inside or on the sides of rectangle "OJ_1DJ_2"[1]. In this area, move along 45° line while not entering a rectangle related to a work center with 1 processor. This diagonal movement is always started from a corner of a rectangle R_j. If located on one side of a rectangle R_j or OJ_1DJ_2, continue your move(s) in that direction to reach the next corner of that side (see Figure 1).

Step 4. Compute the considered performance measure (γ) for all paths from *O* to *D*. Path with the smallest γ value is the optimal sequence. At any path, diagonal moves indicate that J_1 and J_2 are processed simultaneously. However, horizontal or vertical move indicates that only J_1 or J_2 is processed at any time interval by a resource, respectively. For any given path, $\overline{C_w}$ can be computed by using Equation (1).

$$\overline{C_w} = [(x+v)w_1 + (y+h)w_2]/(w_1+w_2) \qquad (1)$$

where, *x* and *y* are the coordinates of *D*; *v* is the sum of vertical moves (except those which are on the right side of OJ_1DJ_2) and *h* is the sum of horizontal moves in a path (except those located on the north side of OJ_1DJ_2). In Figure 1, if $s_1^2 > p_{12}$ and operation is *nr*, one *dummy* extended rectangle will be used for R_1 immediately after e_1^1. This leads to a shift to the other resources rectangles. J_2 isn't processed by R_2, hence, rectangle R_2 is transformed to a horizontal line. This line can be set in a distance $p_{21}+r_2+e_1^1-s_1^1$, e_1^1, s_1^1 and r_2 from axis J_1, too. $C_{\max}$ is equal to *x* in Figure 1. Other performance measures are computed like before.

4. Limited availability problem with set-up times

The proposed algorithm can be extended to the problems with processor (machine) set-up times. The set-up times produce homochromatic boxes with processing time rectangles (transition from both of them is either possible or not). These problems are harder than those without considering set-up times. The proposed method is illustrated in Section 4.1.

4.1. Example 1

Two jobs are to be processed in 4 work centers. All operations are resumeable and all holes are crossable. Job routings and process times are given in the following matrix (numbers in parentheses are p_{ij} and "-" means that J_1 doesn't need work center 4). Other related data are as follows:

$m_1 = m_2 = m_4 = 1$, $m_3 = 2$, $r_1 = 3$,

$r_2 = 2.99$, $S_{11} = 1$, $S_{12} = 2$, $S_{13} = 3$,

$S_{14} = 0$, $S_{21} = 2$, $S_{22} = 0$, $S_{23} = 5$,

$S_{24} = 2$, $w_1 = 1$, $w_2 = 2$, $e_1^1 = 1$,

$e_2^1 = 3$, $e_3^1 = 6$, $e_4^1 = 2$,

$$\forall j : s_j^1 = 0, s_j^2 = 25 \textit{ and } e_j^2 >> 25$$

(">>" indicates: very greater than).

[1] A rectangle made by coordinates axes and points *O* and *D*.

$$\begin{array}{c} \\ J_1 \\ J_1 \end{array} \begin{array}{cccc} R_1 & R_2 & R_3 & R_4 \\ \left[\begin{array}{cccc} 1(3) & 3(1) & 2(3) & - \\ 4(4) & 1(1) & 3(2) & 2(3) \end{array}\right] \end{array}$$

Solution: Geometric graph is shown in Figure 2.

For the shown path, $\overline{C}_w = 20$, $C = 38$, $C_{\max} = 22$ (this path is optimum for makespan and mean total weighted completion time). All set-up time boxes have been aggregated by their processing rectangles. R_4 has a 2-unit vertical line for set-up time and immediately a 3-unit vertical line along with it for processing time. Indeed, length and width of rectangle R_j are $p_{1j} + S_{1j}$ and $p_{2j} + S_{2j}$, respectively. If $m_j = 1$, then the area of rectangle R_j becomes black (impenetrable). Set-up time box can be separated from the processing time rectangle. This has been done for R_3 in Figure 2 and the related Gantt chart is shown in Figure 3. As it can be seen, for $\overline{C}_w$ optimum job sequence is *J*<1-2> for work center 1 and 3, but it is *J*<2-1> for work center 2 (for work center 4 there is only job 2). The dotted (mottle) white rectangles are related to processors unavailability periods and mottle colored rectangles are related to setup times, in Figure 3. After setting up a processor for J_i, the processing of that processor on J_i is started, immediately (see Figure 3). In Example 1, if operations are *nr* or *sr* and/or holes are *ncr* or *cr/ncr*, the results will remain yet the same.

5. Extensions

The proposed method is applicable to resumeable/non-resumeable operations with crossable/non-crossable holes, but it is slightly harder from the view points of computations and drawing. For example in Figure 1, if the first hole on R_1 was non-crossable for J_1 and crossable for J_2, and the operations of R_1 on J_1 were *rs* or *nr* and the operations of R_1 on J_2 were *rs*, first box of R_1 would be transformed to a vertical line with the size of $s_1^1 - r_2$ and second (right) box length of R_2 would be increased to p_{11}. So, the rectangle R_3 was shifted to the right by "$s_1^1 - r_1 - p_{12}$" time unit.

This method is applicable to semi-resumeable operations, too. In this case, set-up times will be added to the continuation of operations after the interruptive hole. It is usually reasonable that the set-up times to be executed completely after the hole, because during the unavailability period, the resource is being repaired (serviced) or utilized for other jobs (events) out of the scheduling system and in result its set-up is changed for the considered job(s). In this status, if the first availability period isn't longer than S_{ij}, setting up the R_j for J_i will be futile. This extension becomes clear by Example 2.

5.1. Example 2

Consider the problem of *FJ*3, $h_{4,j}$, *cr/ncr*| n = 2, *rs/nr/sr*, $r_1 = 3$, $r_2 = 2$, $m_1 = 1$, $m_2 = m_3 = 2$ |C_{max} with set up times in that the operations of "R_1 on J_1 and J_2" and also "R_3 on J_2" are *sr*, the operation of "R_3 on J_1" is *nr* and the operations of "R_2 on J_1 and J_2" are *rs*. Also:

$$\mu_{11} = \mu_{21} = \frac{1}{2}, \qquad \mu_{23} = \frac{1}{3}, \qquad S_{11} = S_{21} = 1,$$

$$S_{12} = S_{22} = S_{13} = 0, \qquad S_{23} = 3, \qquad s_1^1 = 2,$$

$$s_1^2 = 9, \qquad s_2^1 = 17, \qquad s_3^1 = 19, \qquad e_1^1 = 6,$$

$$e_1^2 = 13, \qquad e_2^1 = 18, \qquad e_3^1 = 21.$$

Set-up is completely started after the holes for unfinished *sr* operations. Job routings and process times are given in the following matrix (numbers in parentheses are p_{ij}). Find the optimum solution.

$$\begin{array}{c} \\ J_1 \\ J_1 \end{array} \begin{array}{ccc} R_1 & R_2 & R_3 \\ \left[\begin{array}{ccc} 1(4) & 2(2) & 3(2) \\ 1(3) & 3(1) & 2(1) \end{array}\right] \end{array}$$

Solution: Geometric graph is shown in Figure 4.

The shortest path from *O* to *D* is shown in Figure 4. Optimum $C_{\max}$ value is equal to 29. Note that, the optimal sequence is *J*<2-1> for R_1 and *J*<1-2> for work center 2. It is clear that, setting up is futile for R_3 on J_2, because the operation of R_3 on J_2 is semi-resumeable.

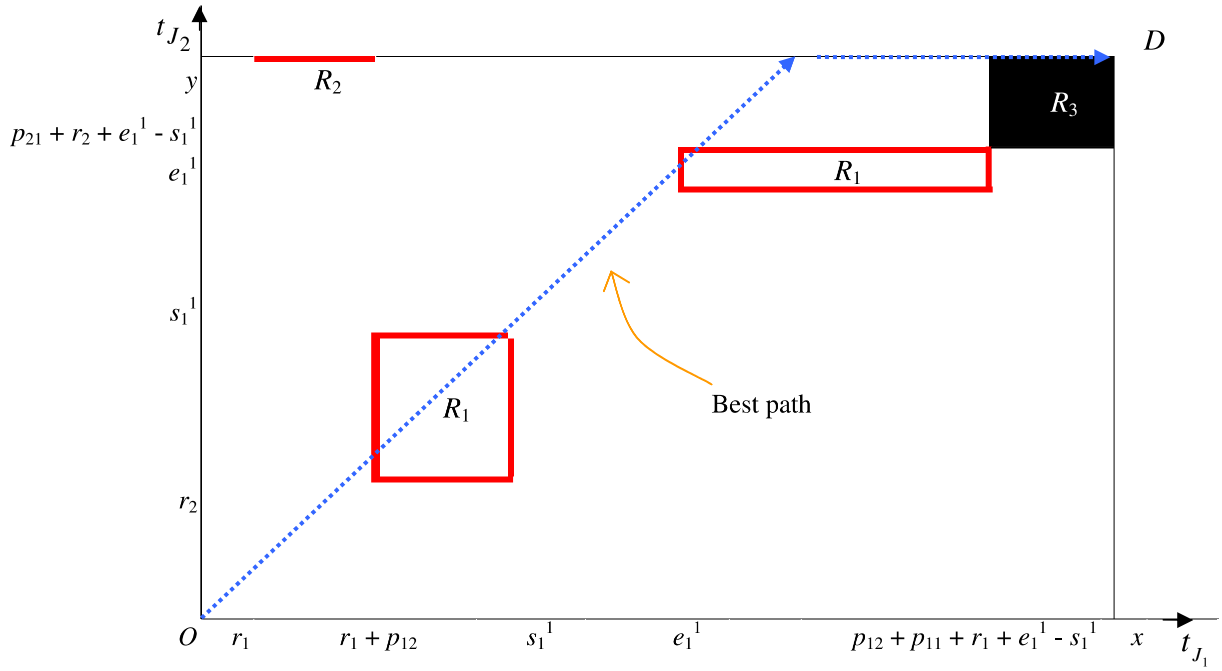

Figure 1. A simple brickwork graph of resumeable job shop.

$$\forall k > 1 : s_1^k > p_{12} + p_{11} + r_1 + e_1^1 - s_1^1,\ e_2^1 < r_1,\ s_2^k > y, e_3^1 < p_{21} + r_2 + e_1^1 - s_1^1,\ s_3^k > x$$

$$x = \sum_{j=1}^{3} p_{ij} + r_1 + e_1^1 - s_1^1,\ y = \sum_{j=1}^{3} p_{2j} + r_2 + e_1^1 - s_1^1,\ m_1 = m_2 = 2,\ m_3 = 1$$

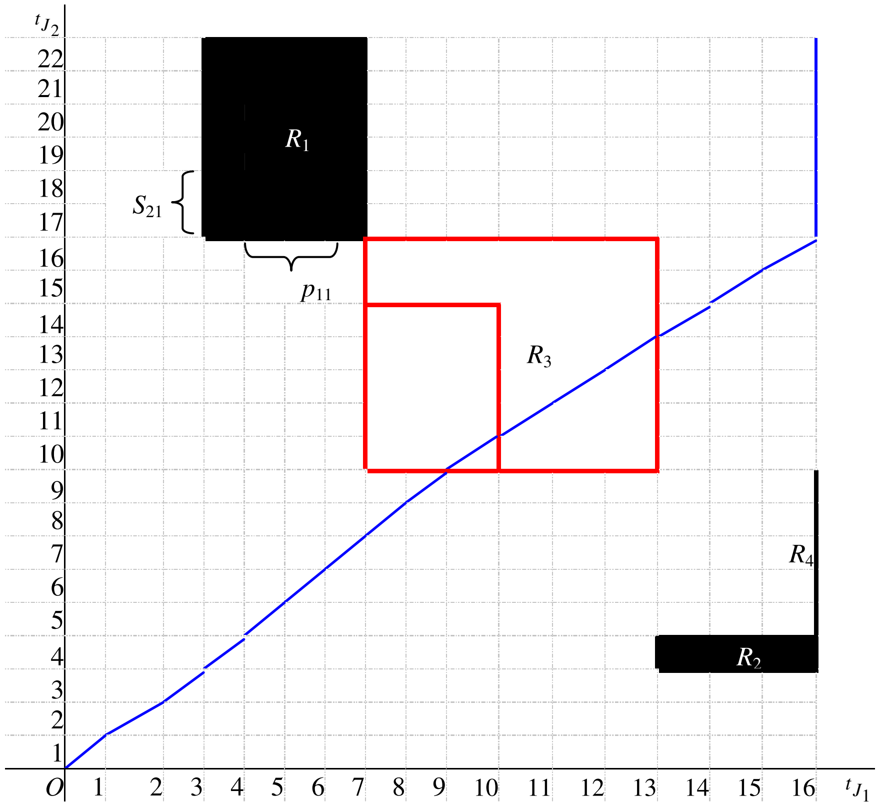

Figure 2. The brickwork graph for Example 1.

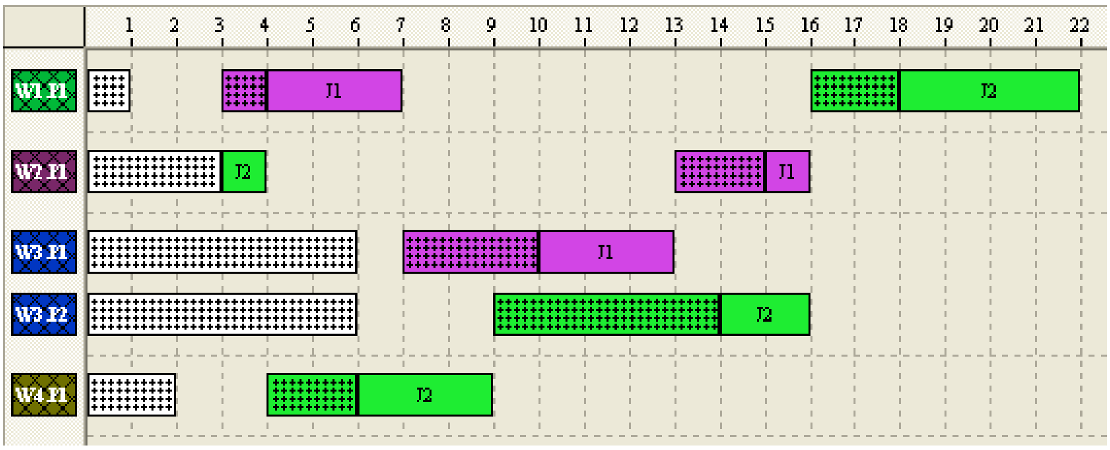

Figure 3. Gantt chart of optimum sequence for Example 1.

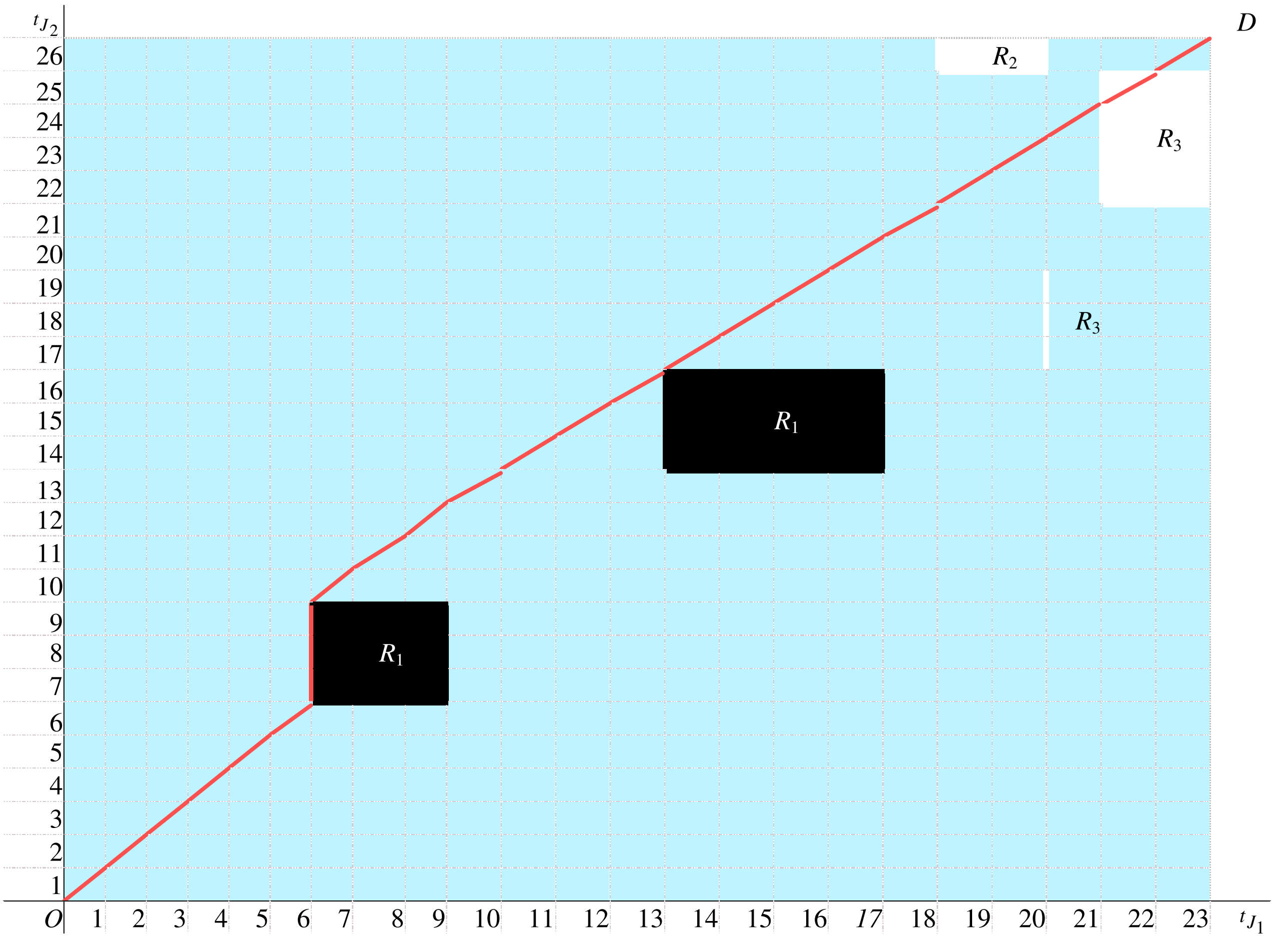

Figure4. Geometric graph of Example 2.

6. Conclusion

We have proposed an exact graphical algorithm based on the extension of Akers' method for solving the problem related to the model FJ, h_{kj} , *cr/ncr*| $n = 2$, *rs/nr/sr*, r_i , $m_j \leq 2 \mid \gamma$, where γ is a performance measure based on the completion time. This algorithm is likely the first algorithm and research work for the resource availability constraint problem in flexible job shop environments. Moreover, the consideration of various environments, arbitrary number of resources (work centers and processors), arbitrary holes on all work centers with any distribution (*nr*, *rs*, *sr*), and ready times are some of the major points considered in our proposed algorithm which have not been propounded in Akers' methods yet."

In continuation, we have described the extension of our algorithm by considering set-up times in the model. Also, the validation and verification of the proposed model were shown by solving a couple of examples. The generalization of Akers' graphical algorithm can be led to extension of above model for arbitrary number of job and m_j . The time of obtaining a solution can be decreased by the transformation of geometric approach to the network technique or any other exact method and heuristics. The proposed algorithm can be extended to n jobs for $C_{\max}$, F, A or any other performance measures too. This extension makes the problem much harder and it is beyond the framework of this paper.

References

[1] Adiri, I., Bruno, J., Frostig, E. and Rinnooy Kan, A.H.G., 1989, Single machine flow-time scheduling with a single breakdown. *Acta Informatica*, 26, 679-696.

[2] Aggoune, R., 2002, *Ordonnancement d'Ateliers sous Contraintes de Disponibilité des Machines*. Ph.D. Thesis, Universite de Metz, France.

[3] Aggoune, R., 2004, *Two-Job Shop Scheduling Problems with Availability Constraints*. Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS 2004), Whistler, Canada.

[4] Akers, S. B. and Friedman, J., 1955, A non-numerical approach to production scheduling problems. *Operations Research*, 3, 429-442.

[5] Balas, E., Lancia, G., Serafini, P. and Vazacopoulos, A., 1998, Job shop scheduling with deadlines. *Journal of Combinatorial Optimization*, 1(4), 329-353.

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

[7] Carlier, J., 1982, The one-machine sequencing problem. *European Journal of Operational Research*, 11, 42-47.

[8] Dehnar Saidy, H. R., 2007, *Job Resources Arrangement Based on Completion Time Criterion.* M.Sc. Thesis, Islamic Azad University (Tehran Sciences and Researches Branch), Iran-Tehran (in Persian).

[9] Kubiak, W., Blażewicz, J., Formanowicz, P., Breit, J. and Schemidt, G., 2002, Two-machine flow shops with limited machine availability. *European Journal of Operational Research*, 136(3), 528-540.

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

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

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

[13] Makui, A., 2000, *Introduction to the Production Planning*. Rozaneh Publication, Tehran, 127-128 (in Persian).

[14] Mauguière, P., Billaut, J. C. and Bouquard, J. L., 2003(a), *Scheduling Resumeable and Non-resumeable Operations*. Proceedings of the Joint International Meeting EURO/INFORMS, Istanbul, Turkey.

[15] Mauguière, P., Billaut, J. C. and Bouquard, J. L., 2005, New single machine and job-shop scheduling problems with availability constraints. *Journal of Scheduling*, 8, 211-231.

[16] Mauguière, P., Bouquard, J. L. and Billaut, J. C., 2003(b), *A Branch and Bound Algorithm for a Job Shop Scheduling Problem with Availability Constraints*. Proceedings of the Sixth Workshop on Models and Algorithms for Planning and Scheduling Problems (MAPSP 2003), Aussois, France.

[17] Mensch, G., 1970, Generalization of Akers' two-dimensional job shop scheduling model to J dimensions. *SIAM Journal on Applied Mathematics*, 18(2), 462-466.

[18] Schmidt, G., 1984, Scheduling on semi-identical processors. *Zeitschrift für Operations Research*, A28, 153-162.

[19] Schmidt, G., 1988, Scheduling independent tasks with deadlines on semi-identical processors. *Journal of Operational Research Society*, 39, 271-277.

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