

Available online at

<http://www.TJMCS.com>

The Journal of Mathematics and Computer Science Vol .2 No.1 (2011) 122-129

Single machine scheduling with fuzzy preemption penalties

Mehdi Heydari

Mheydari@iust.ac.ir

Emran Mohammadi

E_mohammadi@iust.ac.ir

Iran university of science and technology, department of industrial engineering

Received: August 2010, Revised: November 2010

Online Publication: January 2011

Abstract

In preemptive scheduling problems, processing of a job can be temporarily interrupted, and resumed at a later time. Conventionally, in the literature on preemptive scheduling, preempted jobs can simply be resumed from the point at which preemption occurred or restart from the beginning. However, this situation may not always be true in practice. It is likely that, in some cases, an imprecise or fuzzy part of jobs processing must be repeated, i.e., a fuzzy time penalty must be incurred. In this paper, we consider the single-machine scheduling problem of minimizing the total flow time subject to job release dates and fuzzy preemption penalties. We present a heuristic algorithm and validate it using some numerical examples.

Key words

Scheduling; Fuzzy preemption penalty; Membership function; Flow time

1. Introduction

Recently found various researches on the preemptive models of scheduling. Usually for these models, preemptions are assumed to be free, i.e., there isn't any cost for switching to a different

job. However, preemption penalties are normally a full blown fact in practice. There are various overheads in switching to another job. It usually includes setup cost, physically moving jobs on and off a machine, or customers feeling unhappy with their jobs interrupted and delay in completion date. In this paper we consider a preemptive priority scheduling with preemptions subject to fuzzy time penalty, i.e., each time a job is preempted it must restart from a fuzzy given point and other progress will be lost.

Ting and Zheng et al, was studied the preemption-restart model when there are no penalties [1,2]. Their work was a slightly different context known as broadcast scheduling but the results apply here.

Heydari et al considered the online scheduling of jobs in the restart and resume [3]. Also they considered exact setup times as penalties i.e. when a job is preempted, a predetermined setup time must take place. Also Liu and Cheng studied this problem with similar time penalties for preemption [4].

In this paper, we investigate scheduling of single-machine problem for minimizing the total flow time subject to fuzzy time penalty, where each time a job is started after having been preempted, a fuzzy time penalty must take place. This paper is organized as follows. We first present the necessary assumptions and the definitions in section 2. Online preemptive model is introduced in section 3. The paper is supported with a heuristic algorithm in section 4. In section 5 the performance analysis is presented. Finally, conclusion remarks are given in the last part to summarize the contributions of the paper.

2. Necessary assumptions and the definitions

A fuzzy set \tilde{S} in a universe of discourse X is characterized by a membership function $\mu_{\tilde{S}}(x)$ which associate with each element x in X a real number in the interval $[0,1]$. The function value $\mu_{\tilde{S}}(x)$ is termed as the grade of membership of x in \tilde{S} . A triangular fuzzy number \tilde{S} can be defined by a triplet (a_1, a_2, a_3) . Its conceptual schema and mathematical form are shown by following equation:

$$\mu_{\tilde{S}}(x) = \begin{cases} 0 & x < a_1 \\ \frac{x - a_1}{a_2 - a_1} & a_1 < x < a_2 \\ \frac{a_3 - x}{a_3 - a_2} & a_2 < x < a_3 \\ 0 & a_3 < x \end{cases}$$

Also, a triangular fuzzy number \tilde{S} in the universe of discourse X that conforms to this definition is shown in Fig.1. A linguistic variable is a variable the values of which are linguistic terms. Linguistic terms have been found intuitively easy to use in expressing the subjectiveness and/or qualitative imprecision of a decision makers assessments [5].

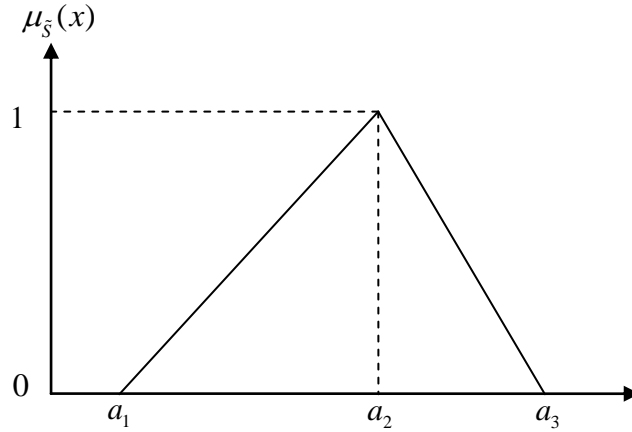


Fig.1. A triangular fuzzy number \tilde{S}

A job J is specified by its processing time (length) t_j and release time r_j , only it can be processed right on or after r_j . Also, we are given a machine that can handle only one job at a time and all jobs may be preempted. The penalty times of job J are denoted by fuzzy number \tilde{S}_j . Completion date for job j is denoted by C_j . Our objective is to find a schedule to minimize the total flow time of the n jobs. It is well known that if $\tilde{S}_j = (0,0,0)$, i.e., no preemption penalty, the above problem is solved by the shortest remaining processing time (*SRPT*) rule: at any time, process the unfinished job with the shortest remaining processing time among the available jobs. However, little is known about the case of $\tilde{S}_j \neq (0,0,0)$.

3. Online preemptive model

Shortest processing time (*SPT*) rule minimize total flow time of n jobs. According to this rule we process the job with the shortest processing time among the available jobs. In an online single machine problem, assume job X with process time t_x is processing on the machine. Let t_{x1} denote the processed time of job X , and, remained processing time is t_{x2} . Note t_x is equal to $(t_{x1} + t_{x2})$. In an online scheduling a new job (Y) with the total processing time t_y and the release time r_y arrives to the shop (Fig.2). Job Y becomes known at its release times.

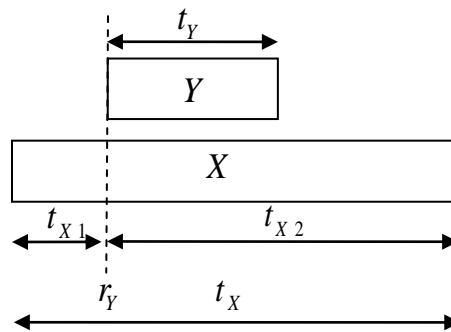


Fig2. Job Y arrives at r_y

In this situation two kind of scheduling is possible. A- If processing of job Y started after completion of job X then according to Fig.3, mean flow time (\bar{F}) is calculated as follows (F_j denote the flow time of job j) :

$$\bar{F} = \frac{F_X + F_Y}{2} = \frac{1}{2}[(t_{X1} + t_{X2}) + (t_{X2} + t_Y)] = \frac{1}{2}[t_{X1} + 2t_{X2} + t_Y]$$

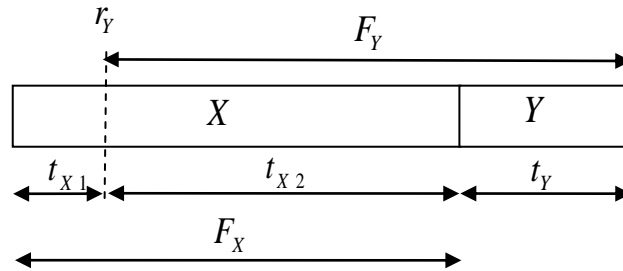


Fig3. Processing of job Y after job X

B-If processing of job Y preferred to job X then according to Fig.4, mean flow time is calculated as follows:

$$\bar{F}' = \frac{F_X + F_Y}{2} = \frac{1}{2}[(t_{X1} + t_Y + (\tilde{S}_j + t_{X2})) + (t_Y)] = \frac{1}{2}[2t_Y + t_{X1} + \tilde{S}_j + t_{X2}]$$

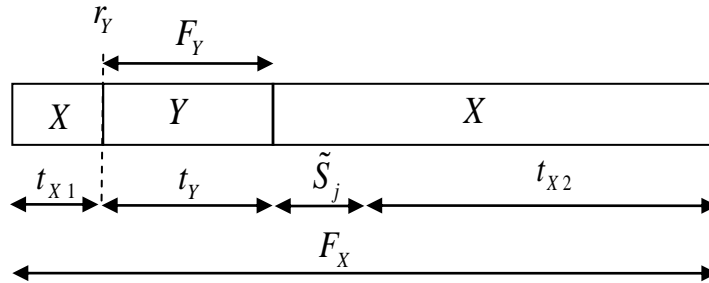


Fig4. Processing of job Y before job X

If $\tilde{S}_j \neq (a^j_1, a^j_2, a^j_3)$ then followings lemma and theorems will be true.

Lemma- In online preemption-restart model, job X is preempted only if $t_Y < t_{X2} - \frac{a^j_1 + 2a^j_2 + a^j_3}{4}$.

Proof: Job X is preempted and job Y is started at its release date only if \bar{F} is less than the \bar{F}' . In the other hand $\bar{F}' - \bar{F} = \frac{1}{2}[t_{X2} - \tilde{S}_j - t_Y] > 0$ or $t_Y < t_{X2} - \tilde{S}_j$ for having a defuzzified result we can replace \tilde{S}_j with $\left(\frac{a^j_1 + 2a^j_2 + a^j_3}{4}\right)$ □.

In order to develop the preemption-restart model one can assume that at zero time, there are n jobs with specified process time in the shop. To minimize the mean flow time we can use the *SPT* rule. Supposing that job Y with release date r_Y arrives to the shop and at this time the machine is processing job X . Let A be the subset of completed jobs and subset B be the non-started jobs (Fig.5). The parameter k is equal to the number of jobs in subset B plus one (The quantity of un-finished jobs).

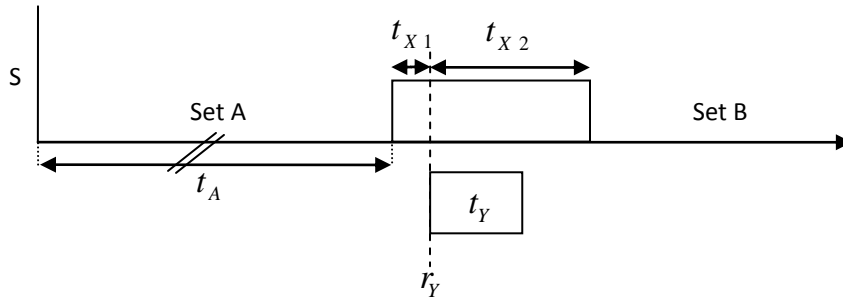


Fig5. Modeling the problem when there are other jobs

Theorem- In online preemption-restart scheduling, job X is preempted if and only if

$$t_Y < t_{X2} - k \times \frac{a_1^j + 2a_2^j + a_3^j}{4}.$$

Proof: Suppose that the preemption conditions are occurred. Therefore, the job X is preempted and the job Y is started at its release dates. In this case, when the job Y is completed, job X will be started and the other job sequencing will not change. Otherwise, if the job X isn't preempted then it will be completed and the *SPT* rule will be used for scheduling un-started works.

According to lemma 1 and lemma 2, $t_Y < t_X$ is the necessary condition for preemption in job X . When this condition holds and according to *SPT* rule, both X and Y have more priority to other jobs. In this situation, one of the following schedules will be optimal.

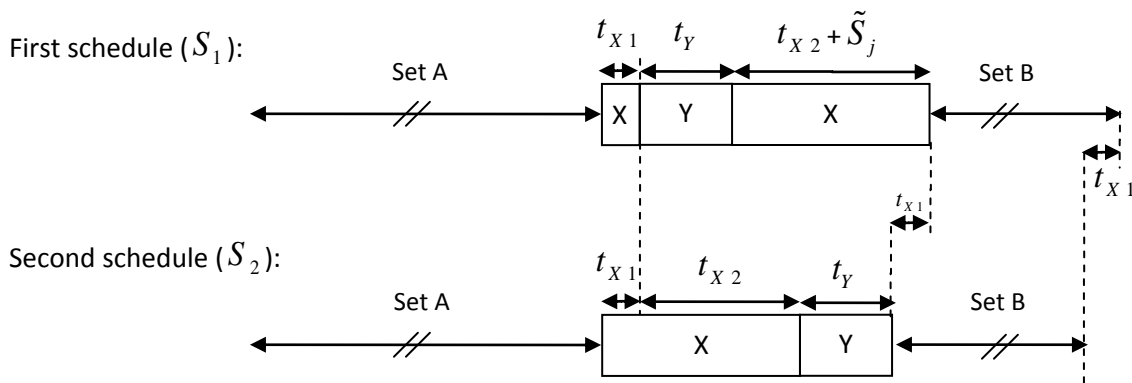


Fig 5. S_1 and S_2

Job X is preempted and job Y is started at its release date only if mean flow time in S_1 is less than mean the flow time in S_2 .

$$\left\{ \begin{array}{l} \bar{F}_2 - \bar{F}_1 = \frac{1}{n+1} \left[\sum_{\forall i \in A} (F_{2i} - F_{1i}) + t_{x2} - t_{x1} - t_Y + \sum_{\forall i \in B} (F_{2i} - F_{1i}) \right] \\ \{ F_{1i} = F_{2i} | \forall i \in A \} \Rightarrow \sum_{\forall i \in A} (F_{2i} - F_{1i}) = 0 \\ \sum_{\forall i \in B} (F_{2i} - F_{1i}) = \sum_{\forall i \in B} (\tilde{S}_j) = (k-1) \times \tilde{S}_j \end{array} \right\}$$

$$\Rightarrow F_2 - F_1 = \frac{1}{n+1} [t_{x2} - k \times \tilde{S}_j - t_Y] \xrightarrow{\text{defuzzification}} F_2 - F_1 = \frac{1}{n+1} \left[t_{x2} - k \times \frac{a_1^j + 2a_2^j + a_3^j}{4} - t_Y \right]$$

Therefore $\bar{F}_2 - \bar{F}_1 > 0$ if and only if $t_Y < t_{x2} - k \times \frac{a_1^j + 2a_2^j + a_3^j}{4}$. □

4- A heuristic algorithm (HM)

Step0-(parameters definition): Suppose i is a counter parameter that its primary content is 1 and the number of jobs is shown with k . Process time of each time is shown with t_i . For each job determine (a_1^j, a_2^j, a_3^j) parameters as follows:

- a_1^j : Optimistic penalty time for preemption of job J
- a_2^j : The most probable penalty time for preemption of job J
- a_3^j : Pessimistic penalty time for preemption of job J

Step1: Determine the priority of each un-finished job according to *SRPT*, from i to $k+i-1$.

Step2: Start processing of job i until its compilation, unless a new job arrives to the shop. If a new job arrives to the shop, dedicate index j to its characteristics and for determining its priority refer to step 4.

Step3: After completion of job i , increase i one unit and decrease k one unit then refer to step1.

Step4: The Processing time for job i before r_j is shown with t_{i1} and the remaining time is shown with $t_{i2} = t_i - t_{i1}$. If problem is preemption-restart go to step7, otherwise follow the algorithm.

Step5: If $t_j < t_{i2} - k \times \frac{a_1^j + 2a_2^j + a_3^j}{4}$ follow the algorithm, otherwise go to step7

Step6: Job i will be preempted and the recently arrived job (job j) will be preferred. From this time index i will be used for recently arrived job. Increase k one unit, and then go to step2.

Step7: Preemption is refused. Continue the processing of job i . Increase k one unit, and then refer to step2.

5-Performance analysis

The proposed algorithm presents practical advantages over other evaluation methods which are done in the past. To examine its results, the proposed method was compared with comparable evaluation methods. The examination depicted that the proposed method always produces more acceptable results for all cases. To evaluate the performance of the proposed heuristic, we have used it on some problems in various sizes and the results are compared with another algorithm developed by Liu and Cheng [8] called H and the next is presented by Heydari et al [3] called HSM. In algorithm H, whenever a job is completed or a new job is released, we must schedule the unfinished job that can be completed at the earliest time (preempting when necessary). Performance of Algorithm HSM is similar to our algorithm except that, only the most probable penalty time is considered.

We have produced 1000 problems in 10 categories where they are formed by different quantities in the number of jobs and mean flow time calculated for these problems. Release dates are generated using uniform distribution in the interval of [0,1000]. The result of comparison is depicted in table-1.

Table-1: Comparing the algorithms subject to mean flow time

# of Jobs	HSM	H	HM
10	53	53	53
30	174	175	173
50	308	312	307
70	476	485	475
90	672	696	669
110	932	998	929
130	1315	1453	1309
150	1861	2240	1858
170	2900	3929	2895
190	5180	7708	5177

As we can observe, the proposed method (HM) performs better than the others when number of jobs increased.

6-Concluding remarks

In this paper, we have studied the single machine scheduling problem of minimizing the total flow time subject to fuzzy preemption penalties, where each time a job is started, after being preempted, a fuzzy setup must take place. Also, a heuristic algorithm is presented and the implementation of the proposed approach is demonstrated using a simulation code. The performance of the proposed method of this paper has been compared with others method and the preliminary results indicate that the proposed method performs better than them.

Scheduling with preemption penalties is a new topic in scheduling research especially for fuzzy problems and we strongly believe that this research could be well extended for

problems with other fuzzy parameters such as fuzzy process time, fuzzy release date, fuzzy due date and so on.

7-References

- 1- Ting H, F (2006) A near optimal scheduler for on-demand data broadcasts. Proc of 6th Italian Conference on Algorithms and Complexity, in: LNCS vol 3998 Springer pp 163–174.
- 2- Zheng F, Fung S.P.Y, Chan W.T, Chin F.Y.L, Poon C.K, Wong P.W.H (2006) Improved on-line broadcast scheduling with deadlines. Proc.of 12th International Computing and Combinatorics Conference, LNCS vol.4112, Springer, pp.320–329.
- 3- Heydari M, Sadjadi S.J, Mohammadi E (2009) Minimizing total flow time subject to preemption penalties in online scheduling. Int J Adv Manuf Technol DOI 10.1007/s00170-009-2190-9.
- 4- Liu Z, Cheng E (2004) Minimizing total completion time subject to job release dates and preemption penalties. Journal of Scheduling 7: 313–327.
- 5- Zadeh L, A (1975) The concept of a linguistic variable and its application approximate reasoning. Information science 8 (3) 199-249.