Journal of the Operations Research
Society of Japan
Vol. 41, No. 4, December 1998

TWO-MACHINES SCHEDULING PROBLEM WITH FUZZY ALLOWABLE TIME CONSTRAINT

Tsutomu Konno Hiroaki Ishii
Osaka University

(Received June 26, 1996; Final February 27, 1998)

Abstract We have already introduced many fuzzy concepts to scheduling problem and discussed so called fuzzy scheduling problems. In this paper, we introduce fuzzy allowable time concept newly to two identical machine problem, which is a fuzzy version of M. R. Garey and D. S. Johnson. That is, there are two identical machines and fuzzy ready times and deadlines are associated to each job, i.e., membership function representing a satisfaction degree of start times and completion times are considered to each job and minimal one of them is to be maximized. Further among jobs, there exists fuzzy precedence relation, which is a fuzzy relation with membership function representing satisfaction degree of precedence order of each job pair and again minimal one of them is to be maximized. The aim is to maximize both minimal degrees at a time if possible, but usually there exists no schedule maximizing both of them, and so we seek nondominated schedules.

1. Introduction

We have already introduced many fuzzy concepts to ordinary scheduling problems and proposed efficient solution procedure for them [3, 4, 5]. In this paper we introduce fuzzy allowable time constraint to the problem considered in [2], i.e., fuzzy ready time and deadline are associated to each job. Corresponding membership functions of each job represent satisfaction degrees of start time and completion time of job processing. Minimal satisfaction degree of them is to be maximized. In a real world, final deadlines depend upon types production priority of job customers etc. For an example, exports are to be completed rigidly before shipping. But in some cases, slight delay is allowed. Further generally speaking, all customers requirements does not necessarily propose to processing his jobs from a start. Further among jobs, there exists a fuzzy precedence relation, which is a fuzzy relation with membership function representing satisfaction degrees of processing order between each job pair and again minimal one of them is to be maximized. Since usually there exists no schedule maximizing both minimal degrees at a time, we seek nondominated schedules.

Section 2 formulates the problem and defines nondominated schedules. Section 3 briefly review the corresponding nondominated scheduling problem in [2]. Section 4 proposes a solution procedure for nondominated schedules of the problem. Section 5 concludes this paper.

2. Formulation of Problem

We consider the following fuzzy scheduling problem. H. Ishii [6] introduced the fuzzy scheduling problem with the fuzzy due-date before. Up to now, no paper about fuzzy scheduling has taken up both fuzzy due-dates and ready time yet.

(1) There are two identical machines and n jobs $J_1, J_2, \cdots, J_n$ that are to be processed by either one of these two machines.

(2) Each job J_i has unit processing time, fuzzy executable start time $\tilde{s}_i$ and fuzzy deadline $\tilde{d}_i$. $\tilde{s}_i$ has the following membership function to represent the satisfaction degree regarding the processing start time s_i of the job (supposed as a nonnegative integer).

$$\mu_{\tilde{s}_i}(s_i) = \begin{cases} 0 & (s_i \leq r_i) \\ m_i(s_i) & (r_i \leq s_i \leq r_i + e_i) \\ 1 & (s_i \geq r_i + e_i) \end{cases}$$

where s_i is integer, and $m_i(s_i)$ is nondecreasing and has a value between 0 and 1. In the similar manner, $\tilde{d}_i$ has the following membership function to represent the satisfaction degree regarding the completion time C_i of the job.

$$\mu_{\tilde{d}_i}(C_i) = \begin{cases} 1 & (C_i \leq d_i) \\ k_i(C_i) & (d_i \leq C_i \leq d_i + f_i) \\ 0 & (C_i \geq d_i + f_i) \end{cases}$$

where, C_i is an integer, and $k_i(C_i)$ is nonincreasing and has a value between 0 and 1. Further $r_i, e_i, f_i, d_i \geq 0, r_i + e_i \leq d_i$, and they are integers. Roughly speaking, $\bar{r}_i = r_i + e_i$ corresponds to the conventional ready time, while d_i corresponds to deadline.

(3) H. Ishii [5] has first introduced a flexible processing order of some pairs of jobs as fuzzy precedence relation. It is defined between two jobs J_i and J_j, and is represented by the satisfaction degree μ_{ij} when J_i precedes J_j, namely, when J_i is processed before J_j. We assume, μ_{ij} is a value between 0 and 1, and, $\mu_{ij} = 1$ in the case of $\mu_{ji} > 0$, while in the case of $\mu_{ij} = \mu_{ji} = 1$, J_i and J_j are independent each order. Further simultaneous processing is not allowed except independent jobs.

(4) When the processing start time of the job J_i under schedule π is represented by s_i^π, and the completion time is represented by C_i^π, then the minimum satisfaction degree $\mu_{r\min}^\pi$ regarding the processing start time in π is given as the follows

$$\mu_{r\min}^\pi = \min \{\mu_{\tilde{s}_i}(s_i^\pi) \mid i = 1, 2, \cdots, n\}.$$

And the minimum value of the satisfaction degree regarding the completion time $\mu_{d\min}^\pi$ is given as follows:

$$\mu_{d\min}^\pi = \min \left\{\mu_{\tilde{d}_1}(C_i^\pi) \mid i = 1, 2, \cdots, n\right\}.$$

Therefore, the satisfaction degree regarding the job execution time μ_1^π is defined as shown below:

$$\mu_1^\pi = \min \{\mu_{r\min}^\pi, \mu_{d\min}^\pi\}.$$

While, the minimum value of satisfaction degree regarding the job processing order μ_2^π is given as:

$$\mu_2^\pi = \min \left\{\mu_{ij} \mid C_i^\pi < C_j^\pi\right\}.$$

(5) Under the above setting, the following problem Q is considered.

$$Q: \quad \mu_1^\pi \rightarrow \max, \ \mu_2^\pi \rightarrow \max \quad \text{subject to } \pi \in \Pi$$

where, Π is a set of all the feasible schedules.

In general, there is no schedule to maximize both μ_1^π and μ_2^π simultaneously, therefore we seek for a nondominated schedule defined next. There is a possibility that there are many nondominated schedules having an identical schedule vector. However, as for the same nondominated schedule vector, only a single schedule of them is sought.

Nondominated Schedule

First of all, schedule vector $\nu^\pi = (\mu_1^\pi, \mu_2^\pi)$ is defined to each schedule π. That the schedule π_1 dominates π_2 means:

$$\nu_1^{\pi_1} \geq \nu_1^{\pi_2}, \quad \nu_2^{\pi_1} \geq \nu_2^{\pi_2}, \quad \nu^{\pi_1} \neq \nu^{\pi_2}$$

for the correspond schedule vectors $\nu^{\pi_1} = (\nu_1^{\pi_1}, \nu_2^{\pi_1})$ and $\nu^{\pi_2} = (\nu_1^{\pi_2}, \nu_2^{\pi_2})$. That the schedule π is called nondominated schedule when there is no schedule dominating π.[3]

3. Nonfuzzy Two-machine Problem with Precedence Relation, Ready Time, and Deadline

Here we briefly review the solution procedure of two-machine problem with nonfuzzy precedence relation, ready time, and deadline by Garey and Johnson [2]. First, the problem is given by the following (1) to (4):

(1) There are two machines and n jobs $J_1, J_2, \cdots, J_n$ that are to be processed by either of these two machines.

(2) Each job J_i has unit processing time, and ready time $\bar{r}_i$ and deadline d_i are defined. Namely, J_i must start processing after $\bar{r}_i$, and complete processing before the deadline d_i.

(3) Precedence relation $\prec$ is defined among some jobs. The expression $J_i \prec J_j$ means that the job J_i must precede the job J_j, in other words, J_i must be processed before the processing of J_j is started. Two jobs not having this relation are called independent each other. To each job J_i, the job J_j in the relation of $J_i \prec J_j$ is called a successor job of J_i.

(4) Under the above setting, a schedule to meet all of ready time, deadline, precedence relation is seeked for.

First, $S(i,s,d)$ to each job J_i and $\bar{r}_i \leq s \leq d_i \leq d$ is defined as a set of all the jobs J_j $(j \neq i)$ that has $d_j \leq d$ and is either the succeeding job of J_i or $\bar{r}_j \geq s$. And $N(i,s,d)$ is defined as the number of its elements. Then when it holds that,

$$N(i,s,d) \geq 2(d-s) \quad \& \quad d - \lceil N(i,s,d)/2 \rceil < d_i$$

deadline of J_i is modified as $d_i = d - \lceil N(i,s,d)/2 \rceil$, where $\lceil \bullet \rceil$ means the minimum integer not below $\bullet$. When this modification is repeated, it leads to either the case where modification is no longer carried out or the case $d_i < \bar{r}_i + 1$ occurs. Algorithm to carry out this modification efficiently is omitted here, but at the moment when correction is not available, it leads to the following internally consistent status. (For details, see [2].)

Internally Consistent

We call the deadlines internally consistent whenever the following conditions hold for every job J_i.

1) $d_i \geq \bar{r}_i + 1$

2) For every pair of integers s, d satisfying $s_i \leq s \leq d_i \leq d$, if $N(i,s,d) \geq 2(d-s)$, then $d_i \leq d - \lceil N(i,s,d)/2 \rceil$.

If it is internally consistent, then $J_i \prec J_j$, implies $d_i \prec d_j$. Jobs are sorted in the order of modified deadline, and numbers are exchanged so that it becomes as $d_i \leq d_{i+1}$, $i = 1, 2, \cdots, n-1$, and priority list $L = (J_1, J_2, \cdots, J_n)$ is constructed. Schedule is made from the head of this list at every moment when some machine become idle checking original precedence relation. Then feasible schedule is obtained. This can be done by $O(n^3)$.

4. Solution Procedure for Fuzzy Version

First sort all μ_{ij} such that $0 < \mu_{ij} < 1$ in the fuzzy precedence relation, and let the results be as shown below:

$$\mu^0 \equiv 1 > \mu^1 > \mu^2 > \cdots > \mu^a > 0$$

where, a is the number of the different μ_{ij}. And, also sort

$$\mu_{\tilde{s}_i}(s_i)\ (r_i \leq s_i \leq r_i + e_i,\ \text{integer}),\ i = 1, 2, \cdots, n$$

and

$$\mu_{\alpha_i}(C_i)\ (d_i \leq C_i \leq d_i + f_i,\ \text{integer}),\ i = 1, 2, \cdots, n$$

and let following be obtained;

$$\mu_0 \equiv 1 > \mu_1 > \mu_2 > \cdots > \mu_b > 0 = \mu_{b+1}$$

where b is the number of different $\mu_{\tilde{s}_i}(s_i), \mu_{\tilde{d}_i}(C_i)$ in $(0, 1)$.

Next, precedence relation graph $PG(V, A)$ is constructed as shown below:

I. V consists of the points ν_i, $i = 1, 2, \cdots, n$ corresponding to each job J_i.

II. The set A of arc is composed of each arc (ν_i, ν_j) when J_i directly precedes J_j.

And the compatible graph $CG(V, A^c)$ is also constructed as below from fuzzy precedence relation: Note, the $CG(V, A^C)$ is undirected graph.

III. V is the point ν_i, $i = 1, 2, \cdots, n$ corresponding to each job J_i.

IV. Edge set A^c is originally composed of edge (ν_i, ν_j) corresponding to independent job pair (J_i, J_j). J_i and J_j can be processed at a time in two identical machines.

Further define arc sets.

$$A^0 = \{(\nu_i, \nu_j) \mid \mu_{ij} = \mu^0, \mu_{ji} \neq \mu^0\}, \quad \bar{A}^l \equiv \{(\nu_i, \nu_j) \mid \mu_{ij} = \mu^0, \mu_{ji} = \mu^l\},\ l = 1, 2, \cdots, a$$

and, the following graphs are defined repeatedly by usual precedence relation graph as follows from these arc set:

$$PG^0 = (V, A^0), \quad A^l = A^{l-1} - \bar{A}^l, \quad PG^l = (V, A^l),\ l = 1, 2, \cdots, a.$$

From these precedence relation graph, ready time, deadline, and precedence relation at that moment, modified deadline is determined, and that is attached to each vertex of compatible graph as a label. To obtain an actual schedule, maximal compatible matching of compatible graph with this modified deadline is obtained as shown below, and it is converted into schedule. Job pair corresponding to each edge that consists of compatible matching are processed at a time. We construct priority list L and schedule is made from the head of L checking the independency not to process the dependent jobs at a time. After the above preparation, the following algorithm is obtained.

[Algorithm]

Step 1: Calculate $\mu^0 \equiv 1 > \mu^1 > \mu^2 > \cdots > \mu^a > 0$ and $\mu_0 \equiv 1 > \mu_1 > \mu_2 > \cdots > \mu_{b+1} \equiv 0$. Set $l = 0$, construct PG^0, and search μ_t, $t = 0, 1, 2, \cdots, b$, set ready time, deadline of each job as $\lceil \mu_{\tilde{s}_1}^{-1}(\mu_t)) \rceil$, $\lfloor \mu_{\alpha_1}^{-1}(\mu_t) \rfloor$, respectively and find the maximum $\mu_t = \mu_{t0}$ to make it feasible, and obtain corresponding schedule π^0. Set, $DS = \{\pi^0\}$, $l = 1$, and go to step 2.

Step 2: Construct PG^l, search μ_t, $t = 0, 1, \cdots, t_{l-1} - 1$, and set ready time, deadline of each job $\lceil \mu_{\tilde{s}_i}^{-1}(\mu_t) \rceil$, $\lfloor \mu_{\alpha_i}^{-1}(\mu_t) \rfloor$, $i = 1, \cdots, n$ respectively find maximum $\mu_t = \mu_{t_l}$ to make it feasible and obtain executable, its schedule π^l. If π^l is not dominated by any schedule of DS, set $DS = DS \cup \{\pi^l\}$, and go to step 3. Otherwise, set $\mu_t = \mu_{t_{l-1}}$ and go to step 3.

Step 3: Set $l = l + 1$. Terminated in the case of $l = a + 1$. Otherwise, return to step 2. (Herein, $\lceil \cdot \rceil$ means the minimum integer not less than the content, and $\lfloor \cdot \rfloor$ means the maximum integer not greater than the content.)

Theorem 1 The above algorithm finds out nondominated schedule in $O((\sum_{i=1}^{n}(f_i + e_i))n^3)$.

Proof: Validity is clear from the above discussion, extension principle of precedence relation to fuzzy precedence relation, and procedure of [2]. Computational complexity is equal to calculation amount of the above result in order since each μ_i, $i = 0, 1, \cdots, b+1$ is checked once, and the check on each feasibility is made in $O(n^3)$ computational time.

5. Conclusion

We extend Garey & Johnson's algorithm for corresponding nonfuzzy problem to fuzzy version and proposed an efficient algorithm. But our above algorithm is straightforward and so may be refined more. The concept of fuzzy allowable time may be introduced into many other scheduling problems. Further, it may be possible to consider a tri-criteria scheduling problem taking fuzzy processing time into consideration such as [7].

References

[1] M. Fujii, T. Kasami and K. Ninomiya: Optimal sequencing of two equivalent processors. *SIAM J. Appl. Math.*, **17** (4) (1969) 784–789.

[2] M.R. Garey and D.S. Johnson: Two-processor scheduling with start-time and deadlines. *SIAM J. Comput.*, **6** (3) (1977) 416–426.

[3] H. Ishii: Two machine Lmax minimization problem with fuzzy precedence constraint. *Proceeding of APORS*, **94** (1994).

[4] H. Ishii and T. Konno: An open shop scheduling with fuzzy allowable time and fuzzy resource constraint. *Fuzzy Sets And Systems* (accept).

[5] H. Ishii and M. Tada: Single machine scheduling with fuzzy precedence relation. *European Journal of Operational Research*, **87** (1995) 284–288.

[6] H. Ishii, M. Tada and T. Masuda: Two scheduling problems with fuzzy duedates. *Fuzzy Sets and Systems*, **46** (1992) 339–347.

[7] M. Tada and H. Ishii: Single machine scheduling with fuzzy precedence relations and fuzzy processing times (to be submitted).

Tsutomu Konno
Faculty of Engineering, Osaka University
Suita, Osaka 565-0871, Japan
E-mail: Tkkonno@aol.com