MACHINE TIME SCHEDULING WITH FUZZY SET OF SCHEDULES[1]

KAREL ZIMMERMANN[2]

ABSTRACT. We consider machine time scheduling problems with crisp release times, processing times and deadlines. The set of feasible schedules is a fuzzy set of schedules which are "acceptable" with respect to a given crisp processing order of jobs. Unlike to [1], [2], [3] the elements of this fuzzy set allow that at most two jobs may be processed simultaneously, no preemption is allowed and schedules with a longer overlapping of processing times have lower membership value. Several membership functions are described and methods for finding a schedule with the maximal membership value are suggested.

1. Introduction, Problem Formulation

Let n operations $1, \ldots, n$ be given, let $p_j > 0$ be the processing time of operation j for $j = 1, \ldots, n$ and s_j the starting time of operation j for $j = 1, \ldots, n$. Let us suppose that no preemption is allowed, so that if operation j starts at time s_j, it is finished (completed) at time $s_j + p_j$. Let $\hat{\mathcal{P}} \equiv \{1, \ldots, n\}$ be a given order, in which the operations $1, \ldots, n$ must be carried out, so that it must be $s_j \geq s_{j-1} + p_{j-1}$ for all $j = 2, \ldots, n$. Let us suppose that each operation j must be started and completed within a given interval $\left[h_j, \tilde{H}_j\right]$, i.e. $[s_j, s_j + p_j,] \subset \left[h_j, \tilde{H}_j\right]$,

[1]Supported by GAČR 201/98/0222
AMS Subject Classification: Primary 15A33 Secondary
Key words: fuzzy scheduling, machine time scheduling
[2]Karel Zimmermann
Department of Applied Mathematics, Faculty of Mathematics and Physics, Charles University Prague, Malostranské nám. 25, CZ-118 00 Prague 1

or equivalently $s_j \in [h_j, H_j]$, where $H_j = \tilde{H}_j - p_j$ for all j. Then the set M of feasible starting-time vectors $s = (s_1, \ldots, s_n)$ is described as follows:

$$s_j \geq s_{j-1} + p_{j-1} \qquad j = 2, \ldots, n \tag{1}$$
$$h_j \leq s_j \leq H_j \qquad j = 1, \ldots, n \tag{2}$$

$M \equiv \{s | s \text{ satisfies (1), (2)}\}$.

It may happen that $M = \emptyset$. The necessary and sufficient conditions under which $M \neq \emptyset$ are given in [5]. The properties of the set M depend on the "input parameters", which are h, H, p, $\hat{\mathcal{P}}$, so that $M = M\left(h, H, p, \hat{\mathcal{P}}\right) = \emptyset$, and a problem can arise how the input parameters h, H, p, $\hat{\mathcal{P}}$ can be changed in such a way that the set of feasible staring times becomes nonempty and the changed parameters are in some sense "close" or "similar" to the original ones. In this context various types of "inverse" problems can be discussed, e.g. if it is allowed to change p, so that it can be $p \in P \equiv \left\{p | \underline{p} \leq p \leq \overline{p}\right\}$, we can accept as an appropriate "approximation" of the empty set $M\left(h, H, \hat{p}, \hat{\mathcal{P}}\right)$ the set $M\left(h, H, p^{\text{opt}}, \hat{\mathcal{P}}\right)$, where p^{opt} is the solution of the problem

$$\|p - \hat{p}\| \equiv \max_{1 \leq j \leq n} |p_j - \hat{p}_j| \to \min$$
$$\text{s.t.} \qquad p \in P \ \& \ M\left(h, H, p, \hat{\mathcal{P}}\right) \neq \emptyset$$

Similarly we could proceed with parameters h, H. Such inverse problems where discussed in [4], [5].

In this contribution, we would like to discuss the possibilities of solving problems, in which $M\left(h, H, p, \hat{\mathcal{P}}\right) = \emptyset$, but h, H, p cannot be changed and $\hat{\mathcal{P}}$ is "recommended" order, which should not be "violated too much". There are of course several approaches how to describe such a "fuzzy" requirement and we shall choose one of them here as a basis for further discussion. The problem consists in choosing appropriate "close" ordering of precessing intervals with respect to $\hat{\mathcal{P}}$. The idea of the approach described here is that the two neighbouring operations $j, j-1$ are allowed to be processed simultaneously (i.e. the time intervals $[s_{j-1}, s_{j-1}+p_{j-1})$ and $[s_j, s_j+p_j]$ may have a nonempty intersection) so that additional machines may be necessary to carry out the operations. If we posed no restrictions to these overlapping of the time intervals, it might happen that in some extreme cases n machines are needed at the same time and the resulting operating order has very little in common with $\hat{\mathcal{P}}$. To avoid this, we allow here only such shifts that

$$s_{j-1} \le s_j \qquad \text{for all} \quad j = 2, \ldots, n$$

and

$$s_{j-2} + p_{j-2} \le s_j \qquad \text{for all} \quad j = 3, \ldots, n$$

This will mean that the starting times are still in the order $\hat{\mathcal{P}} = \{1, \ldots, n\}$, i.e. $s_n \ge s_{n-1} \ge \ldots \ge s_1$ and only two neighbouring operation time intervals $[s_j, s_j+p_j]$, $[s_{j-1}, s_{j-1}+p_{j-1}]$ can overlap in the sense that it holds for all j

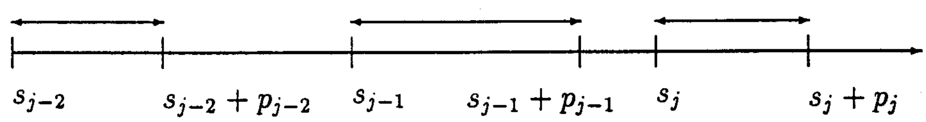
$$s_{j-1} \le s_j \qquad \forall j = 2, \ldots, n \tag{3}$$

$$s_{j-2} + p_{j-2} \le s_j \qquad \forall j = 3, \ldots, n \tag{4}$$

$$\min(s_j + p_j, s_{j-1} + p_{j-1}) - s_j > 0 \qquad \forall j = 2, \ldots, n \tag{5}$$

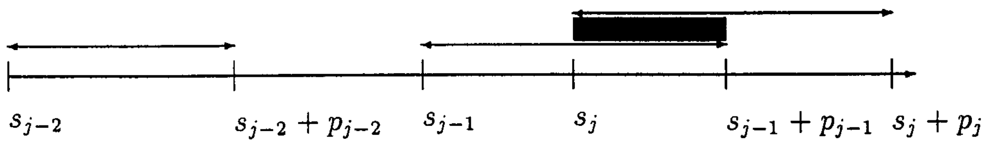
Example 1.1.

(a)



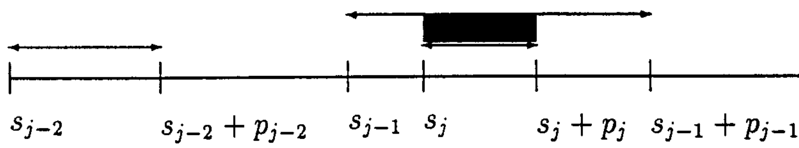
(no overlapping)

(b)



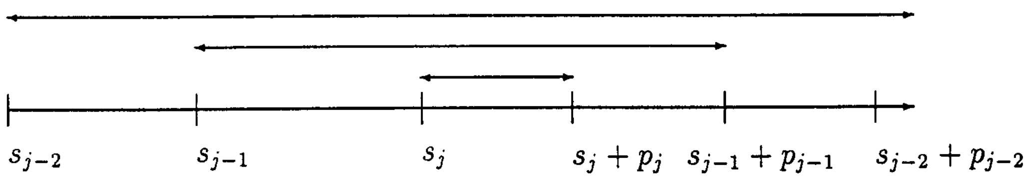
$$\min(s_j + p_j, s_{j-1} + p_{j-1}) - s_j = s_{j-1} + p_{j-1} - s_j > 0$$

(c)



$$\min(s_j + p_j, s_{j-1} + p_{j-1}) - s_j = s_j + p_j - s_j = p_j > 0$$

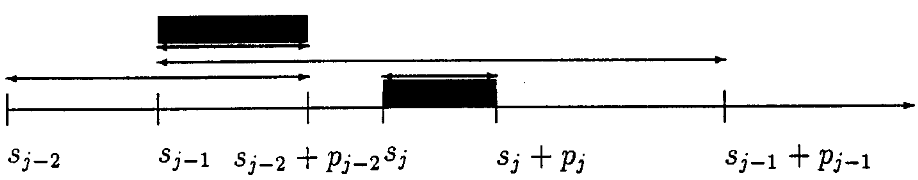
(d)



such overlapping is not allowed, since

$$s_j < s_{j-2} + p_{j-2}$$

(e)



Such overlapping is allowed since

$$s_{j-2} < s_{j-1} < s_j$$
$$s_{j-2} + p_{j-2} < s_j$$

and it is

$$\min(s_{j-1}+p_{j-1}, s_{j-2}+p_{j-2}) - s_{j-1} = s_{j-2}+p_{j-2}-s_{j-1} > 0$$

(overlapping of processing operations $j-1$, $j-2$)

$$\min(s_j+p_j, s_{j-1}+p_{j-1}) - s_j = s_j+p_j-s_j = p_j > 0$$

(overlapping of processing operations j, $j-1$).
In both cases only two operations are carried out at the same time.

We shall define the set $M\left(\hat{\mathcal{P}}\right)$ of feasible vectors of starting times $s = (s_1, \ldots, s_n)$ with respect to order $\hat{\mathcal{P}}$ as follows:

$$M\left(\hat{\mathcal{P}}\right) \equiv \{s | s \text{ satisfies } (2),\ (3), (4)\}$$

Using the same procedure as in [5], [6] we can find out whether the set $M\left(\hat{\mathcal{P}}\right)$ is empty or not. If $M\left(\hat{\mathcal{P}}\right) \neq \emptyset$, we shall choose in some sense "the best element" of $M\left(\hat{\mathcal{P}}\right)$. We can choose as the criterion some measure of overlapping of processing intervals, which will then be minimized on $M\left(\hat{\mathcal{P}}\right)$.

The following three functions can be suggested as measures overlapping of intervals:

$$\varphi^{(i)}(s) = \max_{2\le j\le n} \frac{\varphi^{(i)}(s_{j-1}, s_j)}{\alpha}, \qquad i = 1,2,3 \tag{6}$$

where for $j = 2, \ldots, n$ we define

$$\begin{aligned}
\varphi^{(1)}(s_{j-1}, s_j) &\equiv \max\left(0, \min(s_{j-1}+p_{j-1}, s_j+p_j) - s_j\right) && (7)\\
\varphi^{(2)}(s_{j-1}, s_j) &\equiv \max\left(0, s_{j-1}+p_{j-1}-s_j\right) && (8)\\
\varphi^{(3)}(s_{j-1}, s_j) &\equiv \begin{cases} p_j & \text{if } s_{j-1}+p_{j-1}-s_j > 0 \\ 0 & \text{otherwise} \end{cases} && (9)\\
\alpha &\equiv \max_{1\le j\le n} p_j && (10)
\end{aligned}$$

It is then $\varphi^{(i)} : M\left(\hat{\mathcal{P}}\right) \to [0,1]$ for $i = 1,2,3$ and each of the three measures can be justified by its own "philosophy":

$\varphi^{(1)}$ expresses the measure of simultaneous operating time of two machines;

$\varphi^{(2)}$ expresses the measure of violation of inequality (1);

$\varphi^{(3)}$ expresses the measure of operating time for which second engaged machine must work in case of overlapping.

Remark 1 *Let us remark that we could suggest also other possibilities how to measure the overlapping of time intervals or allow the simultaneous work of more than two machines. It would lead to alternative problems.*

The functions $\varphi^{(i)}$, $i = 1,2,3$ can be interpreted as membership functions of alternative fuzzy sets of "schedules s, which are inappropriate with respect to the given order $\hat{\mathcal{P}}$" so that $1-\varphi^{(i)}(s)$, $i = 1,2,3$, are membership functions of three alternative fuzzy sets of "schedules s, which are appropriate with respect to $\hat{\mathcal{P}}$". The universe of these sets is the set $M\left(\hat{\mathcal{P}}\right)$. In this context we can formulate various problems as for instance:

(a) Find a schedule with maximal membership to the set defined by $1 - \varphi^{(i)}(s)$ (i.e. minimal value of $\varphi^{(i)}(s)$);

(b) Find a schedule $\tilde{s}$ for which $1 - \varphi^{(i)}(\tilde{s}) \geq \beta$ (or $\varphi^{(i)}(\tilde{s}) \leq 1 - \beta$) for a given $\beta \in (0,1)$;

(c) Find a schedule with maximal membership to all the three fuzzy sets (i.e. minimize $\max\left(\varphi^{(1)}(s), \varphi^{(2)}(s), \varphi^{(3)}(s)\right)$

Remark 2 *It is obvious that there are many other problems, which can be formulated in this context. Some of them can be solvable easily, but* *some of them may be even NP-hard. In this contribution, we shall confine only to investigation of the properties of the universal set* $M\left(\hat{\mathcal{P}}\right)$ *and to a method of solving one problem with the membership function* $\varphi^{(2)}$. *Other problems may be the subject of further research.*

Remark 3 *Let us remark that even in case that we allow simultaneous work of two machines it may happen that* $M\left(\hat{\mathcal{P}}\right) = \emptyset$ *and the corresponding fuzzified problems are unsolvable. In this case we can try an approach, which allows simultaneous work of* k *machines for a given* $k > 2$.

Remark 4 *An alternative approach would be to leave the order of completion times unchanged, i.e.* $s_{j-1} + p_{j-1} < s_j + p_j$ *for all* j, *and to relax the order of the starting times* s_j.

Remark 5 *The situation considered in this article may occur if parallel processing of operations is allowed, but it is connected with additional expenses arising for instance from the necesity of additional machines or workers. If there is no feasible schedule satisfying the given requirements without parallel processing, the feasiblity without parallel processing can be achieved only if we change some of the input parameters* h_j, H_j *or* p_j. *It may happen that the shortening of processing times* p_j *is impossible because of technological reasons and we cannot also change the time intervals* $[h_j, H_j]$ *(e.g. in case that the operations produce unstable or perisable items that must be available in prescribed time periods). In such situations, we can proceed in such a way that we allow parallel processing of operations, but because of economic reasons we try to minimize the time duration of the parallel processing. In the sequel, only two operations are allowed to be processed at the same time.*

2. Properties of the Feasible set $M\left(\hat{\mathcal{P}}\right)$

Let $M\left(\hat{\mathcal{P}}\right)$ be defined as in the preceding section. We shall derive a procedure, which enables us to find out whether $M\left(\hat{\mathcal{P}}\right) \neq \emptyset$ and if it is nonempty, to find its greatest element $\overline{s}$ in the sense that

$$\overline{s} \in M\left(\hat{\mathcal{P}}\right) \qquad \text{and} \qquad s \leq \overline{s} \quad \forall s \in M\left(\hat{\mathcal{P}}\right).$$

Let us set at the beginning $\overline{s} \equiv H$. If $H \in M\left(\hat{\mathcal{P}}\right)$, then $M\left(\hat{\mathcal{P}}\right) \neq \emptyset$ and $\overline{s} = H$ is its greatest element. If $\overline{s} = H \notin M\left(\hat{\mathcal{P}}\right)$, we shall proceed as follows: Since any element of $M\left(\hat{\mathcal{P}}\right)$ must satisfy the inequalities $s_1 \leq s_2 \leq \ldots \leq s_n$, the time $\overline{s}_n = H_n$ is the latest possible starting time for operation n and it holds at the same time that

$$s_{n-1} \leq s_n \leq H_n$$

and

$$s_{n-2} + p_{n-2} \leq s_n$$

so that it must be $\underline{s}_n \equiv \max(h_{n-1}, h_{n-2} + p_{n-2}, h_n) \leq \overline{s}_n \equiv H_n$. If this inequality is not fulfilled, then $M\left(\hat{\mathcal{P}}\right) = \emptyset$. Otherwise we accept $\overline{s}_n = H_n$ as the latest possible starting time for operation n and proceed further to s_{n-1}. It must hold:

$$\begin{aligned} h_{n-1} &\leq s_{n-1} \leq \min(H_{n-1}, \overline{s}_n) \\ s_{n-2} &\leq s_{n-1} \\ s_{n-3} + p_{n-3} &\leq s_{n-1} \end{aligned}$$

so that

$$\underline{s}_{n-1} \equiv \max(h_{n-1}, h_{n-2}, h_{n-3} + p_{n-3}) \leq s_{n-1} \leq \min(H_{n-1}, \overline{s}_n)$$

If $\max(h_{n-1}, h_{n-2}, h_{n-3}+p_{n-3}) \leq \min(H_{n-1}, \overline{s}_n)$ we accept $\overline{s}_{n-1} \equiv \min(H_{n-1}, \overline{s}_n)$ as the latest possible starting time for operation $n-1$; otherwise $M\left(\hat{\mathcal{P}}\right) = \emptyset$. Let us consider still s_{n-2}, it must be

$$\begin{aligned} h_{n-2} &\leq s_{n-2} \leq H_{n-2} \\ s_{n-3} &\leq s_{n-2} \\ h_{n-4} + p_{n-4} &\leq s_{n-2} \leq \overline{s}_{n-1} \end{aligned}$$

so that

$$\underline{s}_{n-2} \equiv \max(h_{n-2}, h_{n-3}, h_{n-4} + p_{n-4}) \leq s_{n-2} \leq \min(H_{n-2}, \overline{s}_{n-1})$$

If $\max(h_{n-2}, h_{n-3}, h_{n-4}+p_{n-4}) > \min(H_{n-2}, \overline{s}_{n-1})$, then $M\left(\hat{\mathcal{P}}\right) = \emptyset$, otherwise we accept $\overline{s}_{n-2} \equiv \min(\overline{s}_{n-1}, H_{n-2})$ as the latest possible starting time for operation $n-2$ and proceed further. We obtain in general

$$\begin{aligned} \underline{s}_{n-k} &\equiv \max(h_{n-k}, h_{n-k-1}, h_{n-k-2} + p_{n-k-2}) \leq s_{n-k} \leq \overline{s}_{n-k} \equiv \\ &\equiv \min_{0 \leq j \leq k} H_{n-k+j} \qquad \text{for } k = 0, \ldots, n-3 \end{aligned} \tag{11}$$

$$\underline{s}_2 \equiv \max(h_1, h_2) \leq s_2 \leq \overline{s}_2 \equiv \min_{2 \leq j \leq n} H_j \tag{12}$$

$$\underline{s}_1 \equiv h_1 \leq s_1 \leq \overline{s}_1 \equiv \min_{1 \leq j \leq n} H_j \tag{13}$$

Therefore if $M\left(\hat{\mathcal{P}}\right) \neq \emptyset$, there exist always $\underline{s}$, $\overline{s} \in M\left(\hat{\mathcal{P}}\right)$ satisfying (11)-(13) and for any $s \in M\left(\hat{\mathcal{P}}\right)$ it is $\underline{s} \leq s \leq \overline{s}$. It can be easily verified that this procedure either interrupts on a step k, $0 \leq k \leq n-1$ with $\max(h_{n-k}, h_{n-k-1}), h_{n-k-2} + p_{n-k-2} > \min(H_{n-k}, \overline{s}_{n-k-1}$ and the answer that $M\left(\hat{\mathcal{P}}\right) = \emptyset$, it gives us in the step $k = n-1$ the greatest and the smallest element of $M\left(\hat{\mathcal{P}}\right)$ and since $\underline{s}$, $\overline{s} \in M\left(\hat{\mathcal{P}}\right)$, it is $M\left(\hat{\mathcal{P}}\right) \neq \emptyset$.

3. Optimization Problems

We shall assume in this section that $M\left(\hat{\mathcal{P}}\right) \neq \emptyset$ so that according to the results of the previous section there exist elements $\underline{s}$, $\overline{s} \in M\left(\hat{\mathcal{P}}\right)$ with the property $\underline{s} \leq s \leq \overline{s}$ for any $s \in M\left(\hat{\mathcal{P}}\right)$. Let us consider now the following optimization problem:

$$\left.\begin{array}{l} \varphi^{(2)}(s) \equiv \max\limits_{2 \leq j \leq n} \varphi_j^{(2)}(s_{j-1}, s_j) \rightarrow \min \\ \text{s.t.} \quad s \in M\left(\hat{\mathcal{P}}\right) \end{array}\right\} \tag{14}$$

where $\varphi_j^{(2)}(s_{j-1}, s_j)$ is defined as in (8).

It is evident that for any $\alpha \geq 0$ we have

$$\varphi^{(2)}(s) \leq \alpha \Leftrightarrow \varphi_j^{(2)}(s_{j-1}, s_j) \leq \alpha \quad \forall j$$

and for any $j \in \{2, \ldots, n\}$

$$\varphi_j^{(2)}(s_{j-1}, s_j) \leq \alpha \Leftrightarrow s_{j-1} + p_{j-1} - s_j \leq \alpha$$

i.e.

$$s_{j-1} + p_{j-1} - \alpha \leq s_j \tag{15}$$

We can replace now condition (2) with condition, which takes into account both (2) and (15), i.e.

$$\max(s_{j-1} + p_{j-1} - \alpha, s_{j-1}) \leq s_j \tag{16}$$

Let us set

$$M\left(\hat{\mathcal{P}}, \alpha\right) \equiv \{s | s \text{ satisfies } (16), (3), (4)\}$$

Using the same procedure as in the previous section, we can find out whether $M\left(\hat{\mathcal{P}}, \alpha\right) \neq \emptyset$. This can be used to employ a binary search procedure for finding an approximate ε-optimal solution of (14), i.e. a schedule $s^{(\varepsilon)} \in M\left(\hat{\mathcal{P}}, \alpha^{(\varepsilon)}\right)$, where $\varphi^{(2)}(s^{(\varepsilon)}) \leq \alpha^{(\varepsilon)})$ and $\varphi^{(2)}(s^{(\varepsilon)}) - \varphi^{(2)}(s^{\mathrm{opt}}) \leq \varepsilon$, where $\varepsilon > 0$ is a given accuracy and s^{opt} is the optimal solution of (15).

Remark 6 *There exist also finite procedures for obtaining $\alpha^{\mathrm{opt}} = \varphi^{(2)}(s^{\mathrm{opt}})$. One possibility is to empty a subgradient feasible direction procedure starting with $s^{(0)} \equiv \overline{s}$ and decreasing on each step only "active" $\varphi_j^{(2)}(s^{(0)})$, which are equal to $\varphi^{(2)}(s^{(0)})$. Another possibility is to reformulate the problem a linear programming problem and employ a special solution method adjusted to the corresponding linear programming problem with a special structure.*

Remark 7 *The optimal solution of (14) gives us a schedule from the set $M\left(\hat{\mathcal{P}}\right)$ with maximal membership value to the fuzzy set of schedules "appropriate with respect to $\hat{\mathcal{P}}$"; the fuzzy set is described by the membership function $\varphi^{(2)}$.*

References

[1] Ishii, H., Tada, M.: Single MAchine Scheduling Problem with Fuzzy Precedence Relation, European Journal of OR 87, pp.284-288.

[2] Levner, E., Vlach, M.: Single Machine Scheduling with Fuzzy Precedence Constraints, Research Report IS-RR-97-0031F, Japan Advanced Institute of Science and Technology, Tatsunokuchi, 1997.

[3] Tanaka, K., Vlach, M.: Single Machine Scheduling with Fuzzy due Dates, in Proceedings of IFSA'97 Part I, pp.30-35, Academia, Prague, 1997.

[4] Tharwat, A., Zimmermann, K.: On Optimal Choice of Parameters in Machine Time Scheduling Problems, Proceedings "Mathematical Methods in Economy and Industry", Liberec 1998, ISBN 80-7083-338-6, pp.107-112.

[5] Vlach, M., Zimmermann, K.: Optimal Starting Times for a Class of Single Machine Scheduling Problems with Earliness and Tardiness Penalties, Kybernetika, vol.33, No.6, 1997, pp.675-686.

[6] Vlach, M., Zimmermann, K.: Machine Time Scheduling: Synchronization of Starting Times, Proceedings of Conference Mathematical Methods in Economics 1999, ISBN 80-7079-371-6, pp.277-281, Jinřichův Hradec, 1999.