Journal of the Operations Research
Society of Japan
Vol. 42, No. 3, September 1999

SINGLE-MACHINE SCHEDULING WITH MIXED PRECEDENCE CONSTRAINTS

Eugene Levner
Center for Technological Education Holon

Milan Vlach
Japan Advanced Institute of Science and Technology

(Received July 8, 1998; Revised January 29, 1999)

Abstract The paper deals with a single machine scheduling problem involving a general precedence structure that permits both ordinary and fuzzy precedence constraints. Feasible schedules are evaluated not only by their cost but also by the degree of satisfaction with their precedence structure. An $O(m \log n + \max(n^2, kn^2))$-time algorithm is proposed for finding nondominated solutions of the resulting bi-criteria scheduling problem where n is the number of jobs, k is the number of fuzzy constraints, and m is the total number of precedence constraints.

1. Introduction

There are many scheduling problems of practical interest in which the input data are uncertain or imprecise, and their forms of uncertainty or imprecision are not of probabilistic nature. In situations where objective functions or constraints are neither deterministic nor probabilistic, the problems may often be modelled with the help of fuzzy sets.

The aim of this paper is to study a scheduling problem involving both ordinary (crisp) and fuzzy precedence constraints. Machine scheduling problems with such constraints may occur in many different situations in modern automated systems. We now present several examples of such problems.

Consider a multi-machine robotic cell in which a robot (an automated guided vehicle) is responsible for loading/unloading and transporting parts between machines. The parts visit the machines according to given routing tables, thus generating demand for the robot. Various possible equipment layouts in such cells are presented, for example, in [1,8,12,13]. The due dates of robot operations within a production cycle are specified by a process engineer, according to specific requirements of a production process. Further, the technological requirements impose that certain robot operations must precede some others. The decision maker is faced with the problem of finding a robot's schedule subject to precedence constraints and minimizing some cost function depending on completion times of robot operations.

Another example relates to a pick-and-place robot serving an automated warehouse. Suppose n items (orders) are waiting on their places in the warehouse for being sent to the customers. The pick-and-place robot has to serve them in turn, that is, to unload and transport them to the output one after another. Due dates and unloading/transportation times are known for each item. Some items may be unloaded only after some others, and these precedence constraints on robot's operations are specified in advance. Again, the decision maker is to find a schedule for the robot such that the precedence constraints are satisfied and a given function of operation completion times is minimized.

Our next example relates to the mission planning of a satellite (a space robot) [3,4].

Consider a satellite following orbits around the Earth in order to take shots corresponding to images requested by various customers. One of the decision problems to be solved is the so-called daily task planning which consists in sequencing of tasks (shots) to be performed by the satellite during a planning period. Due dates and operation durations are known for each task. Precedence constraints between the tasks that range them according to their scientific, military, political or commercial importance are determined by experts and specified in advance. Since all the images requested cannot usually be satisfied in time, a penalty function is given for each task the value of which depends on time when the task is performed. In order to observe the interests of all customers, a shot sequence minimizing the maximum penalty (or some other function of penalties) is to be found. Often the quality of the resulting shot sequence must be evaluated by taking into account this criterion as well as the satisfaction of the decision maker with the task ordering.

In all examples given above, the precedence constraints may be crisp or fuzzy. The crisp precedence constraints describe the ordering between the operations which is not liable to any change under any circumstances (for example, it is a given technological order) while the fuzzy precedence constraints describe a more flexible situation where some pairs of operations are not linked by technological constraints but the decision maker prefers one ordering to another according to her or his intuition or experience. The choice of an order among these two possible alternatives for each pair of such operations is, in fact, a decision variable. As we shall see below, this choice may be carried out in such a way that the satisfaction of the decision maker with a schedule is maximized. The exact meaning of this criteria will be explained in the next section.

In the case when all precedence constraints are crisp, we obtain the classical single-machine n-job scheduling problem for which a great number of results is reported in the scheduling literature. Depending on the specific form of the objective functions and constraints of the problem, many versions of the problem are known to be polynomially solvable while a lot of others are proved to be NP-hard (see, for example, the extensive survey [10]). However, not too much is known about this scheduling problem when fuzzy precedence constraints are present.

Recently, Ishii and Tada [6] have addressed a single machine problem of minimizing maximum lateness in the presence of fuzzy precedence constraints. They proposed an efficient combinatorial algorithm solving the problem in cubic time (with the problem size being measured in the number of jobs).

In this paper, we extend the above model in the following two aspects. Firstly, we deal with minimizing the maximum of arbitrary nondecreasing cost functions of completion times. Secondly, we consider a rather general precedence structure which includes both crisp and fuzzy precedence constraints.

Following the Ishii-Tada approach, we take the degree of satisfaction of the decision maker with respect to the precedence structure as the second criterion, and propose an $O(m \log n + \max(n^2, kn^2))$-time algorithm for finding nondominated solutions of this bi-criteria scheduling problem, where n is the number of jobs, k is the number of fuzzy constrains and m is the total number of crisp and fuzzy precedence constraints.

In the next section we present the problem formulation. In Section 3 we present a new strongly polynomial algorithm for finding nondominated solutions of the resulting bi-criteria scheduling problem. In Section 4 we analyze the algorithm and obtain, as 'by-products', refinements of the Lawler and Ishii-Tada algorithms. Section 5 presents a numerical example. Section 6 summarizes the results of the paper and discusses some directions for further research.

2. Problem Formulation

We consider the following scheduling problem with *mixed* (crisp and fuzzy) precedence constraints.

1. There are n jobs $J_1, \ldots, J_n$ to be processed on a continuously available machine which can execute at most one job at a time.
2. No preemption is permitted, i.e. each job J_j requires an uninterrupted processing time p_j, $j = 1, \ldots, n$, where all p_j are positive rational numbers.
3. Two types of precedence constraints between the jobs are given:
   (a) Crisp constraints that are specified in the form of a partial order given by an acyclic digraph $G = (V, E)$, each node $j \in V$ corresponding to job J_j, and each arc $(j, k) \in E$ depicting that job J_j has to be completed before the processing of job J_k can start.
   (b) Fuzzy constraints that impose that some pairs of jobs can be carried out in an arbitrary order but a certain order is more preferable than the opposite order, from decision maker's point of view.

   Thus, for each unordered pair of different jobs, J_i and J_j, exactly one of the following three mutually exclusive options occur: either (i) J_i and J_j are subject to a crisp constraint, or (ii) J_i and J_j are subject to a fuzzy constraint, or (iii) J_i and J_j are independent, that is, are related neither by a crisp nor by a fuzzy constraint.

   Each fuzzy precedence constraint is specified by (i) an unordered pair of different jobs, say $U = \{J_i, J_j\}$ and (ii) a membership function μ_U defined on the corresponding two-element set consisting of ordered pairs (J_i, J_j) and (J_j, J_i). Here (J_i, J_j) denotes that J_i precedes J_j.

   To simplify the notation, we denote the two values of μ_U by μ_{ij} and μ_{ji}, that is, $\mu_{ij} = \mu_U(J_i, J_j), \mu_{ji} = \mu_U(J_j, J_i)$. The values μ_{ij} and μ_{ji} express the degrees of satisfaction when job J_i precedes job J_j and when J_j precedes J_i, respectively. We assume that each membership function μ_U is normal and unimodal, that is, one of the numbers μ_{ij} and μ_{ji} is 1, and the other is strictly between 0 and 1.
4. Associated with each job J_j is a nondecreasing cost function f_j; if job J_j is completed at time C_j , cost $f_j(C_j)$ is incurred. Special cases of the general cost function are, for example, the lateness and tardiness.

   Throughout the paper we denote the set $\{1, 2, \ldots, n\}$ of job indices by J and identify schedules with permutations of J. In other words, we are optimizing over the set of permutation schedules. We say that a permutation π of J is *compatible* with a directed graph whose node-set is J, if, for each arc (i, j) from the transitive closure of the graph, job J_i precedes J_j (not necessarily immediately) in permutation π.

   When evaluating the quality of a schedule, we wish to take into consideration not only the cost of the schedule but also the degree of satisfaction with the precedence structure of the schedule. As a consequence, we obtain a bicriteria scheduling problem. One objective is to minimize the maximum cost, and the other is to maximize the minimum degree of satisfaction with the job order. Since μ_{ij} and μ_{ji} are given for the fuzzy constraints only, we first define μ_{ij} and μ_{ji} for the remaining ordered pairs of jobs by setting

   - $\mu_{ij} = 1$ and $\mu_{ji} = 0$ for each pair (i, j) belonging to the transitive closure of the crisp precedence graph G;
   - $\mu_{ij} = \mu_{ji} = 1$ for each pair of independent jobs J_i and J_j.

For each permutation $\pi = (\pi(1), \pi(2), \ldots, \pi(n))$ of $\{1, 2, \ldots, n\}$ that is compatible with all crisp constraints, we define

$$\mu_{min}(\pi) = \min_{1 \le i < k \le n} \mu_{\pi(i),\pi(k)} \tag{2.1}$$

$$f_{max}(\pi) = \max_{1 \le j \le n} f_j(C_j(\pi)) \tag{2.2}$$

where $C_j(\pi)$ denotes completion time of job J_j under the permutation schedule given by π.

Since it may happen that no feasible schedule optimizes f_{max} and μ_{min} simultaneously, we are interested in finding schedules which are nondominated with respect to the following dominance relation: A schedule π_1 is said to be *dominated* by a schedule π_2 if

$$f_{max}(\pi_1) \ge f_{max}(\pi_2), \;\; \mu_{min}(\pi_1) \le \mu_{min}(\pi_2),$$

and at least one of these two inequalities is strict.

If π and π' are such that

$$f_{max}(\pi) = f_{max}(\pi'), \quad \mu_{min}(\pi) = \mu_{min}(\pi')$$

then we call them *equivalent.* Consequently, every *equivalence class* of schedules consists of all schedules with identical value of f_{max} and identical value of μ_{min}.

Now the problem we are addressing in this paper can be formulated in the following way.

Problem P: Find one schedule from each equivalence class of nondominated feasible schedules.

Let us note that this problem includes several known scheduling problems as special cases. For example:

(a) In the case that only crisp precedence relation is imposed, the above problem reduces to the Lawler single-machine scheduling problem [9].

(b) If the set of crisp precedence constraints is empty, and each cost function f_j is the lateness with respect to a given due date d_j that is, $f_j(C_j) = C_j - d_j$, then the above problem reduces to the Ishii-Tada problem with fuzzy precedence relation [6].

(c) If precedence relations (both crisp and fuzzy) are absent and functions f_j are specified as membership functions measuring the degree of expert's dissatisfaction, then the problem becomes a scheduling problem with fuzzy due dates (see, for example, Ishii et al.[7], Han et al. [5], Tanaka and Vlach [14], and Vlach [15]).

3. Problem Analysis and Algorithm

In what follows, G_α stands for a digraph defined for each $0 < \alpha \le 1$ as follows: If $\alpha = 1$, then $G_\alpha = G_1$ is the digraph obtained from the crisp precedence graph G by adding to it all arcs (i, j) corresponding to the fuzzy constraints with $\mu_{ij} = 1$. If $0 < \alpha < 1$, then G_α is the digraph obtained from G_1 by deleting all arcs (i, j) such that $\mu_{ij} = 1$ and $\mu_{ji} \ge \alpha$. After this deletion, there remain only arcs (i, j) with $\mu_{ij} = 1$ and $\mu_{ji} < \alpha$ in the resulting graph. Note that graph G_α may contain directed cycles for some values of α.

For ease of description of solution procedures for problem **P**, we consider also the following auxiliary problems $P_\alpha, 0 < \alpha \le 1$:

$$\begin{array}{ll} \text{Minimize} & f_{\max}(\pi) \\ \text{subject to} & \mu_{min}(\pi) \geq \alpha, \\ & \pi \text{ is compatible with } G_\alpha. \end{array}$$

As pointed out in the previous section, Ishii and Tada [6] proposed an $O(n^3)$-time algorithm for solving problem **P** in the case that no crisp constraints are present, graph G_1 is acyclic, and f_{max} is the maximum lateness. For reader's convenience we recall the basic steps of their algorithm.

THE ISHII-TADA ALGORITHM

Step 1 [Initialization]

1.1 Arrange all μ_{ij} from the open interval $(0, 1)$ in nondecreasing order and rename the resulting k different values so that

$$1 > \mu(1) > \mu(2) > \cdots > \mu(k) > 0.$$

1.2 Set $q := 0,\ \mu(q) := 1,\ \alpha := \mu(q),\ L := \emptyset$.

1.3 Construct G_α as described above.

Step 2 [Solving P_α] Minimize the maximum lateness subject to the precedence relation given by G_α. If the resulting solution π_α is not dominated by any schedule from L and π_α is not yet in L, then include it in L.

Step 3 [Arc deletion] If $q < k$, then set $q := q + 1$, $\alpha := \mu(q)$, construct G_α by deleting from $G_{\mu(q-1)}$ all arcs (i, j) with $\mu_{ji} = \alpha$, and go to Step 2. If $q = k$, then stop (the current L provides the required set of nondominated solutions).

The extension of the Ishii-Tada approach to the general case proposed in this paper is based on the following four observations.

Observation 1 Solution of problem P_α in the second step of the Ishii-Tada algorithm can be implemented in polynomial time not only for the maximum lateness but also for an arbitrary function f_{max} defined by (2), provided all f_j are nondecreasing and their values can be computed in constant time.

Proof. The well known algorithm of Lawler [9] provides the required possibility. To obtain a concise description of this algorithm we use the following notation. Let the precedence relation under consideration be given by an acyclic digraph $G = (V, E)$. Without loss of generality we assume that V is the set of the job indices $\{1, 2, \ldots, n\}$. For each subset I of V, let G_I be the digraph obtained by restricting G to I. Furthermore, for each $i \in I$, let $T_i(G_I)$ denote the set of all nodes in I preceded (not necessarily immediately) by node i, and let $E_i(G_I)$ denote the set of all arcs of G_I entering node i. Now we are ready to describe the algorithm.

THE LAWLER ALGORITHM

$I := V,\ A := E,\ u := \sum_{i \in I} p_i$
For $j = n$ down to 1 do:
  Choose $i \in I$ such that
  $f_i(u) = \min_{\{j \in I \mid T_j(G_I) = \emptyset\}} f_j(u)$
  $I := I \setminus \{i\}, A := A \setminus E_i(G_I)$
  $u := u - p_i$
  $\pi(j) := i$
output (π)

The algorithm determines the index $\pi(i)$ of the job in the i-th position of an optimal permutation schedule, and the analysis presented in [9] shows that the algorithm can be run in $O(n^2)$-time. □

Observation 2 If graph G_α contains a cycle, then problem **P** has no feasible solution π with $\mu_{min}(\pi) \geq \alpha$.

Proof (by contradiction). Assume that problem **P** has a feasible solution π with $\mu_{\min}(\pi) \geq \alpha$, and that graph G_α has a directed cycle C. Rename the nodes of V so that $\pi = (1, 2, \ldots, n)$. Let $C = (v_1, v_2, \ldots, v_t, v_1)$ where $t \leq n$. Let u denote the smallest node in C, and w the node standing just before u in C. [For instance, if $\pi = (1,2,3,4,5,6)$ and $C = (6,5,3,6)$ then $u = 3$ and $w = 5$]. Consider two possible cases.

CASE A. Suppose that arc (w, u) of C (in graph G_α) corresponds to a crisp constraint.

According to the definition of the crisp constraint, it means that job J_w must precede job J_u in all feasible schedules. On the other hand, any feasible permutation (and π as well) determines uniquely an order of processing the jobs. Since, in accordance with our choice, $u < w$ in π, then J_u is to be processed before J_w in schedule π. This condition, taken together with the above one that J_w precedes J_u, contradicts our basic assumption that the crisp precedence graph $G = (V, E)$ is acyclic.

CASE B. Arc (w, u) in G_α corresponds to a fuzzy constraint.

According to the construction of G_1 and G_α, $\mu_{w,u} = 1$. However, J_u precedes J_w in π (because of $u < w$). Therefore, μ_{uw} will be participating in defining the value $\mu_{min}(\pi)$ according to (1). Recall now that graph G_α is constructed by deleting from G_1 all arcs with $\mu_{ji} \geq \alpha$, leaving only those (i, j) for which $\mu_{ji} < \alpha$. Then $\mu_{min}(\pi) < \alpha$, which contradicts the assumption that $\mu_{\min}(\pi) \geq \alpha$. This proves the claim. □

Observation 3 If graph G_β contains a cycle for some $0 < \beta \leq 1$, then the maximum value of α in the interval $0 < \alpha < \beta$ for which G_α is acyclic can be found by a binary search.

Proof. First we notice that if problems P_{α_1} and P_{α_2} with $0 < \alpha_1 < \alpha_2 \leq \beta$ are such that the corresponding graphs G_{α_1} and G_{α_2} have cycles, then also the graph G_α corresponding to P_α contains a cycle whenever $0 < \alpha_1 < \alpha < \alpha_2$. This is obvious because G_{α_1} is a subgraph of G_α for each $0 < \alpha_1 < \alpha < \beta$. It remains to show that there exists α in the interval $(0, \beta)$ such that G_α is acyclic. According to the assumption, graph G_β contains a cycle. It follows that G_β contains arcs corresponding to fuzzy constraints, because the graph defining the crisp constraints is assumed to be acyclic. It follows, also from the acyclicity of the crisp precedence, that we can obtain an acyclic graph G_α with $0 < \alpha < \beta$ by deleting from G_β sufficiently many arcs corresponding to fuzzy constraints. Now, if graph G_β contains a cycle we can arrange all μ_{ij} values from $(0, 1)$ corresponding to its fuzzy constraints in decreasing order. Obviously the number of different μ_{ij} values is not greater than m, the total number of all precedence constraints. Since graph G is an acyclic subgraph of G_β, the standard binary search will identify the maximal μ_{ij} for which the graph $G_{\mu_{ij}}$ is also acyclic in $O(\log n)$ time. □

Observation 4 If G_β is acyclic for some $0 < \beta \leq 1$, then there is a generalization of the arc deletion procedure of the third step of the Ishii-Tada algorithm such that it can be applied for a transition from P_β to P_α with $0 < \alpha < \beta$.

Proof. Consider an arbitrary feasible schedule π for problem P_α with $0 < \alpha < \beta$. Let jobs J_i and J_j be such that

- the arc (i, j) belongs to G_β,
- $\mu_{ij} = 1$ and $\mu_{ji} \geq \alpha$.

If job J_i is scheduled before job J_j in π, then there are k and l such that

$$k < l, \quad \pi(k) = i, \quad \pi(l) = j.$$

If job J_i is scheduled after job J_j in π, then there are k and l such that

$$k < l, \quad \pi(k) = j, \quad \pi(l) = i.$$

It follows that in both cases $\mu_{\pi(k)\pi(l)} \geq \alpha$. However, as Ishii and Tada noticed in [6], this implies that an arc (i, j) can be discarded from G_β without violating the feasibility condition $\mu_{min}(\pi) \geq \alpha$. Moreover, the minimum of f_{max} subject to the precedence constraints given by $G_\beta \backslash \{(i, j)\}$ cannot be smaller than the minimum of f_{max} subject to G_β, because $G_\beta \backslash \{(i, j)\}$ is a subgraph of G_β. This argument can be repeated iteratively for all arcs (i, j) of G_β for which $\mu_{ij} = 1$ and $\mu_{ji} \geq \alpha$, as required by the Ishii-Tada arc deletion procedure. □

Algorithm Description

Step 1 [Initialization]

1.1 Arrange all μ_{ij} from the open interval $(0, 1)$ in nondecreasing order and rename the resulting k different values so that

$$1 > \mu(1) > \mu(2) > \cdots > \mu(k) > 0.$$

1.2 Set $q := 0, \ \mu(q) := 1, \ \alpha := \mu(q), \ L := \emptyset$.

1.3 Construct G_α by adding to the crisp precedence graph all arcs (i, j) corresponding to the fuzzy constraints with $\mu_{ij} = 1$.

Step 2. Check if graph G_α is acyclic. If yes, go to Step 4; if no, go to Step 3.

Step 3. [Binary search] Find i such that $\mu(i)$ is the greatest value among $\mu(1), \ldots, \mu(k)$ such that the corresponding graph $G_{\mu(i)}$ is acyclic. Set $\alpha := \mu(i)$.

Step 4. [Solving P_α] Solve problem P_α. If the resulting solution π_α is not dominated by any schedule from L and π_α is not yet in L, then include it in L.

Step 5 [Arc deletion] If $q < k$, then set $q := q + 1$, $\alpha := \mu(q)$, construct G_α by deleting from $G_{\mu(q-1)}$ all arcs (i, j) with $\mu_{ji} = \mu(q - 1)$, and go to Step 4. If $q = k$, then stop (the current L provides the required set of nondominated solutions).

4. Analysis of the Algorithm

In what follows we will assume, without loss of generality, that all fuzzy arcs have different membership function values. If some arcs have identical μ_{ij} values, for example, $\mu(1) = \mu(2) = \ldots = \mu(d)$, then we let $\mu'(i) = \mu(i) + \epsilon i$, for $i = 1, \ldots, d$, where $\epsilon > 0$ is a small number and we use ϵi to break ties.

Theorem *The time complexity of the algorithm is $O(m \log n + \max(n^2, kn^2))$ where k is the number of fuzzy constraints, m is the total number of precedence constraints, and n is the number of jobs.*

Proof.

Step 1.1 requires $O(k \log n)$ operations.

Step 1.2 and Step 1.3 can be run in $O(m)$ time.

Step 2 can be done in $O(m)$ time (see, for example [2]).

Step 3 is run no more than $\log m$ times, each time requiring $O(m)$ operations.

Step 4 requires $O(n^2)$, according to the complexity of the Lawler algorithm [9] when the scheduling problem is solved from scratch. Then this step is repeated at most $k+1$ times, each time requiring $O(n^2)$ operations. Thus the total time for this step is $O(kn^2)$.

Step 5 requires at most $O(m)$ time.

Thus, the overall time complexity of the algorithm is $O(m \log n + \max(n^2, kn^2))$. □

Corollary 1. (*Refinement of the Lawler-Moore algorithm for* $L_{\max}$). The time complexity of minimizing the maximum lateness subject to crisp precedence only is $O(m + n \log n)$.

Proof. Note that $O(m)$ time is needed to construct graph TG, the topologically sorted representation of the directed acyclic graph G [2]. Let $1, 2, \ldots, n$ be the new numbers of nodes of G, linearly ordered in TG.

According to the Lawler-Moore algorithm [11] for minimizing the maximum lateness $L_{\max} = \max_j (C_j - d_j)$, we need to determine and sort the modified due dates:

$$d'_j = \min\{d_j, \min\{d_i | J_i \in T_j\}\},\ j = 1, \ldots, n,$$

where T_j is the set of all successors (not necessarily immediate) of j in TG.

Let W_j be the set of immediate successors of j in TG and N_j be the number of nodes in W_j. Since TG is topologically sorted, we can rewrite the definition of d'_j as follows:

$$d'_j = \min\{d_j, \min\{d_i | J_i \in W_j\}\},\ j = 1, \ldots, n.$$

Let us calculate $d'_n, d'_{n-1}, \ldots, d'_1$ in this order. In order to know d'_n we now need $O(1)$ time. Further, in order to know d'_{n-1} we need $O(N_{n-1})$ operations, to know d'_{n-2} we need $O(N_{n-2})$ operations, and so on, till we find d'_1 using $O(N_1)$ operations.

The total time is $O(N_1 + \cdots + N_{n-1}) = O(m)$.

To sort the d'_j, we need $O(n \log n)$ time, and this proves the claim. □

The estimation refines the Lawler-Moore result [11] in practically important cases when m is less than n^2, and coincides with it in the case when m is close to n^2, for example, if G is complete.

Corollary 2. (*Refinement of the Ishii-Tada algorithm*). If G_1 is an acyclic digraph in which the maximal distance between pairs of its nodes is H, then the algorithm can be run in $O(m \log n + Hk)$ time.

Proof. When applying the above algorithm to the Ishii-Tada problem, Steps 2 and 3 are not exploited, while Step 4 requires now only $O(m)$ operations when the problem is solved from scratch. In addition, $O(Hk)$ time is needed, because $O(H)$ operations are applied to recalculate the modified due dates, the latter recalculations being repeated $O(k)$ times, each time for a new problem P_α. □

The estimation refines the Ishii-Tada result in [6] when m is less than n^2, and coincides with it when m is $O(n^2)$. As an obvious consequence of this corollary, we obtain the following result.

Corollary 3. (*The case of planar graphs*). If, in addition to the assumption of the previous Corollary, the precedence graph is planar, then the problem can be solved in $O(n \log n + Hn)$ time.

5. Example

Consider the following 5-job instance of our problem specified as follows.

Processing times:

$$p_1 = 2,\ p_2 = 5,\ p_3 = 2,\ p_4 = 3,\ p_5 = 4.$$

Crisp precedence constraints (constituting graph G):

$$(J_1 \text{ precedes } J_4),\ (J_2 \text{ precedes } J_4) \text{ and } (J_3 \text{ precedes } J_4).$$

Fuzzy precedence constraints:

$$\begin{array}{lllll} \{J_1, J_2\}, & \{J_1, J_3\}, & \{J_2, J_3\}, & \{J_2, J_5\}, & \{J_4, J_5\}, \\ \mu_{12} = 1, & \mu_{13} = 1, & \mu_{23} = 1, & \mu_{25} = 0.9, & \mu_{45} = 1, \\ \mu_{21} = 0.95, & \mu_{31} = 0.8, & \mu_{32} = 0.7, & \mu_{52} = 1, & \mu_{54} = 0.75. \end{array}$$

Cost functions:

$$f_i(t) = \begin{cases} 0 & \text{if } t \le \underline{d}_i, \\ (t - \underline{d}_i)/(\overline{d}_i - \underline{d}_i) & \text{if } \underline{d}_i < t \le \overline{d}_i, \\ 1 & \text{if } \overline{d}_i < t, \end{cases}$$

where $\underline{d}_i$ and $\overline{d}_i$ are given by

$$\begin{array}{lllll} \underline{d}_1 = 20,, & \underline{d}_2 = 40,, & \underline{d}_3 = 50,, & \underline{d}_4 = 65,, & \underline{d}_5 = 40. \\ \overline{d}_1 = 150, & \overline{d}_2 = 100, & \overline{d}_3 = 120, & \overline{d}_4 = 120, & \overline{d}_5 = 80. \end{array}$$

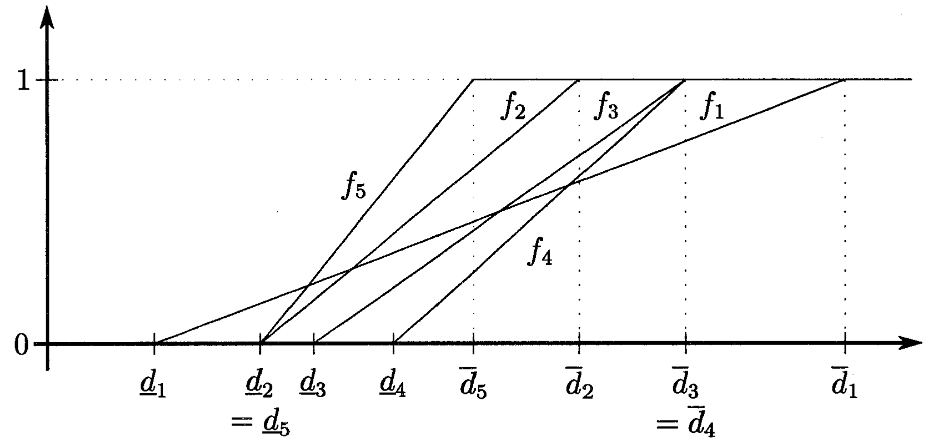

Figure 1: Cost functions

Obviously these cost functions (see Figure 1) can be interpreted as fuzzy due dates which describe the degree of dissatisfaction of a decision maker with job completion times as follows. The decision maker is completely satisfied (his degree of dissatisfaction is zero) when job J_i is completed by time $\underline{d}_i$, and his degree of dissatisfaction increases linearly as the job completion time increases, achieving the maximal value 1, if the job is completed at or after $\overline{d}_i$ (see for example [5,7,14,15]).

The proposed algorithm works as follows.

Step 1: We obtain $k = 5$ and

$$1 = \mu(0) > \mu(1) > \cdots > \mu(5) > 0$$

where $\mu(1) = \mu_{21}$=0.95; $\mu(2) = \mu_{25} = 0.9$; $\mu(3) = \mu_{31} = 0.8$; $\mu(4) = \mu_{54} = 0.75$ and $\mu(5) = \mu_{32}$=0.7. Also graph G_1 is constructed (see graph (a) of Figure 2).

Step 2: It turns out that graph G_1 contains a cycle. Thus, we must go to Step 3.

Step 3: Step 3 is performed in three iterations.

In the first iteration, we find that graph $G_{0.8}$ is acyclic. In the second iteration, we find that graph $G_{0.9}$ is also acyclic. Finally in the third iteration, we find that $G_{0.95}$ has a cycle. After the third iteration we go to Step 4.

Step 4: Step 4 is performed sequentially four times, for $\alpha = 0.9, 0.8, 0.75$ and 0.7, respectively (see Figure 3).

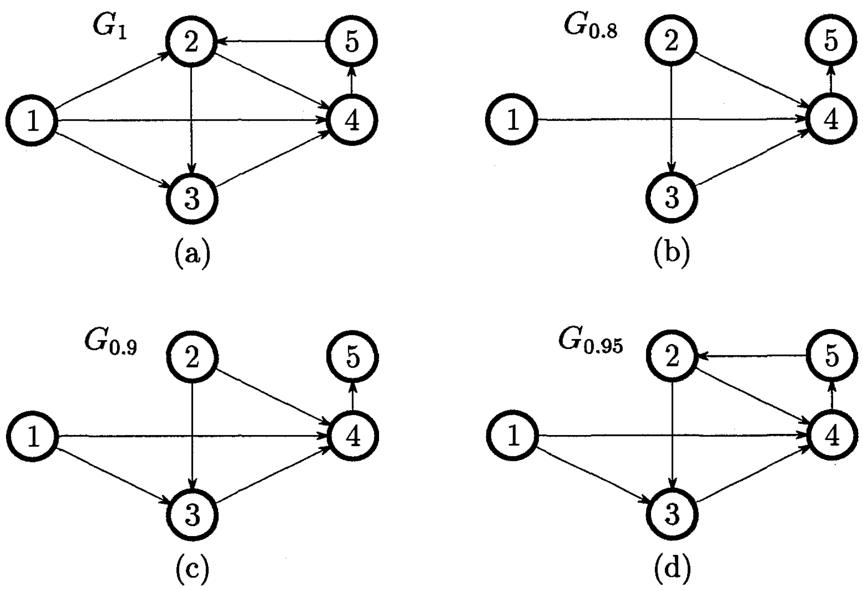

Figure 2: Binary search of an acyclic graph

At each iteration of Step 4, the corresponding problem P_α is solved with the help of Lawler's algorithm. The corresponding solutions are:

if $\alpha = 0.9$ then $\pi_1 = (1, 2, 3, 4, 5)$ with $f_{\max}(\pi_1) = 1$ and $\mu_{\min}(\pi_1) = 0.9$;
if $\alpha = 0.8$ then $\pi_2 = (2, 3, 1, 4, 5)$ with $f_{\max}(\pi_2) = 1$ and $\mu_{\min}(\pi_2) = 0.8$;
if $\alpha = 0.75$ then $\pi_3 = (5, 2, 3, 1, 4)$ with $f_{\max}(\pi_3) = 0.3$ and $\mu_{\min}(\pi_3) = 0.75$;
if $\alpha = 0.7$ then $\pi_4 = (5, 3, 2, 1, 4)$ with $f_{\max}(\pi_4) = 0.25$ and $\mu_{\min}(\pi_4) = 0.7$.

Notice that schedule π_2 is discarded at Step 4 because it is dominated by schedule π_1. Three remaining schedules constitute the set of nondominated solutions, for this example.

6. Concluding Remarks

This paper introduces a new practically important class of one-machine scheduling problems, the so-called problems with mixed precedence constraints, which include both crisp and fuzzy precedences. It is shown that the two types of precedences not only have different physical nature but also require a different algorithmic treatment in the scheduling problems. The scheduling problem under consideration is formulated as a bi-criteria combinatorial optimization problem, and a set of nondominated solutions is obtained in strongly polynomial time. Examples of applications in robotic automated systems are presented.

Several extensions of this study may be of practical value. First, it is certainly possible that the polynomial algorithm presented in this work can be further improved.

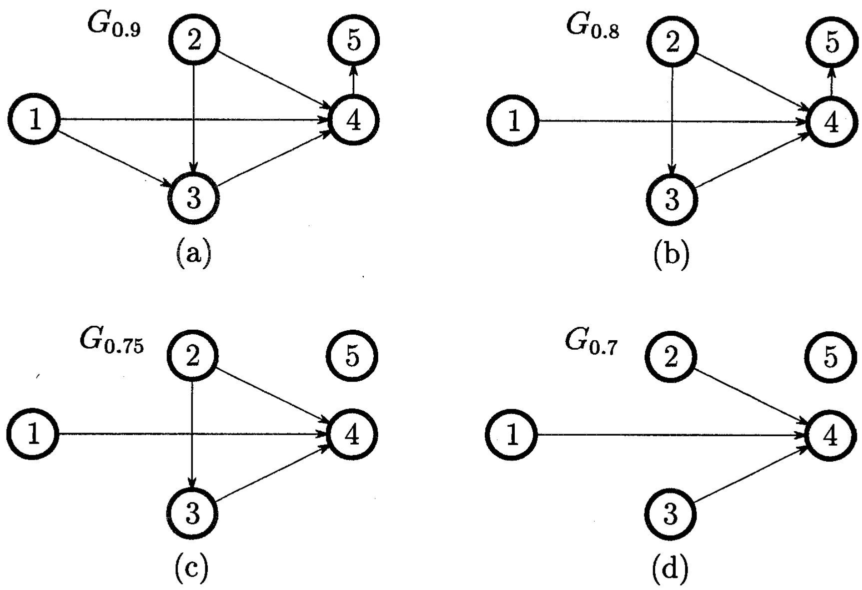

Figure 3: Solution process

Another extension is concerned with applying the suggested approach for fuzzy scheduling in wider practice-oriented directions: (a) more than one machine may be treated with other criteria and scheduling scenarios, like job shop, open shop, cyclic shop or general shop, (b) other types of fuzzy constraints and fuzzy input data could be involved, (c) synchronization and coordinated scheduling of machines and automated material-handling devices (robots) in flexible manufacturing systems are of special interest.

Acknowledgments

This work was performed while the first author was visiting the Japan Advanced Institute of Science and Technology, Ishikawa. The support of this author by the International Information Science Foundation (grant 97.1.3.615) and by the Ministry of Science of Israel in the framework of the cooperative program with the Japan Society for the Promotion of Science (grant 8951.1.97) is gratefully acknowledged. The authors thank J.K. Lenstra for a very helpful comment.

References

[1] J.L. Cheng, H. Kise, and Y. Karuno: Optimal scheduling for an automated m-machine flowshop. *Journal of Operations Research Society of Japan*, **40**-3 (1997) 1-15.

[2] T.H. Cormen, C.E. Leiserson, and R.L. Rivest: *Introduction to Algorithms* (The MIT Press, Cambridge, 1990).

[3] C. Gaspin: Mission scheduling. *Telematics and Informatics*, **6** (1989) 159-169.

[4] N.G. Hall and M.J. Magazine: Maximizing the value of a space mission. *European Journal of Operational Research*, **78** (1994) 224-241.

[5] S. Han, H. Ishii, and S. Fujii: One machine scheduling problem with fuzzy duedates. *European Journal of Operational Research*, **79** (1994) 1-12.

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

[7] H. Ishii, M. Tada, and T. Masuda: Two scheduling problems with fuzzy due-dates. *Fuzzy Sets and Systems*, **46** (1992) 339-347.

[8] H. Kise, T.Shioyama, and T. Ibaraki: Automated two-machine flowshop scheduling: a solvable case. *IIE Trans.*, **23** (1991) 10-16.

[9] E.L. Lawler: Optimal sequencing of a single machine subject to precedence constraints. *Management Science*, **19** (1973) 544-546.

[10] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys: Sequencing and scheduling: Algorithms and complexity. In S.Graves, A. Rinnooy Kan and P. Zipkin (eds.): *Logistics of Production and Inventory, Handbooks in Operations Research and Management Science* (Elsevier, New York, 1993), Vol.4, 445-522.

[11] E.L. Lawler and J.M. Moore: A functional equation and its application to resource allocation and scheduling problems. *Management Science*, **16** (1969) 77-84.

[12] Y.H. Pao and M. Jelinek: Flexible Manufacturing Cells and Systems. In R.C. Dorf and S.Y. Nof (eds.): *International Encyclopedia of Robots: Application and Automation* (Wiley, New York, 1988), 530-551.

[13] E. Levner, K. Kogan, and I. Levin: Scheduling a two-machine robotic cell: A solvable case. *Annals of Operations Research*, **57** (1995) 217-232.

[14] K.Tanaka and M.Vlach: Single machine scheduling with fuzzy due dates. In: *Proceedings of the VII International Fuzzy Systems Association World Congress, IFSA'97* (Academia, Prague, 1997), Vol. I, 30-35.

[15] M. Vlach: Scheduling and sequencing in a fuzzy environment. In: *Proceedings of the VII International Fuzzy Systems Association World Congress, IFSA'97* (Academia, Prague, 1997), Vol. III, 195-199.

Milan Vlach
School of Information Science,
Japan Advanced Institute of Science and Technology,
1-1 Asahidai, Tatsunokuchi, Ishikawa 923-1292, Japan.
E-mail: `vlach@jaist.ac.jp`