Fuzzy constraints in job-shop scheduling[1]

Didier Dubois – Hélène Fargier – Henri Prade

Institut de Recherche en Informatique de Toulouse (I.R.I.T.) – C.N.R.S.
Université Paul Sabatier, 118 route de Narbonne
31062 Toulouse Cedex, France

Abstract: This paper proposes an extension of the constraint-based approach to job-shop scheduling, that accounts for the flexibility of temporal constraints and the uncertainty of operation durations. The set of solutions to a problem is viewed as a fuzzy set whose membership function reflects preference. This membership function is obtained by an egalitarist aggregation of local constraint-satisfaction levels. Uncertainty is qualitatively described is terms of possibility distributions. The paper formulates a simple mathematical model of jobshop scheduling under preference and uncertainty, relating it to the formal framework of constraint-satisfaction problems in Artificial Intelligence. A combinatorial search method that solves the problem is outlined, including fuzzy extensions of well-known look-ahead schemes.

1. Introduction

There are traditionally three kinds of approaches to jobshop scheduling problems: priority rules, combinatorial optimization and constraint analysis. The first kind of method has the merit of being computationally very efficient and can be easily applied to real world cases. Surveys on performance analysis of priority rules can be found in Blackstone et al. (1982), Montazeri (1990), and Grabot and Geneste (1994). Using priority rules, there is no guarantee as to the quality of the obtained solution, especially if some temporal constraints should be respected. It is only known that some of them perform better than others on the average. The optimization methods (see Bellman et al, 1982, for instance) are much more rigorous but are not tractable in large size problems, if the optimal solution is what is searched for. Some progress in efficiency is expected from stochastic optimization methods (e.g., Laguna et al., 1991; Van Laarhoven et al., 1992). Anyway, optimizing a single criterion (e.g., minimizing the completion time of the last job) leads to a very limited view of the real problem. An optimal solution in the mathematical sense is not always useful at the practical level due to unmodelled criteria. The third approach, initiated by Erschler et al.(1976), looks for a set of feasible solutions that obey several temporal or technological constraints, leaving the choice of the final solution to the user. More recently the knowledge-based scheduling school (Fox and Zweben, 1993) has proposed to tackle the jobshop scheduling problem in terms of constraint-directed search methods stemming from Artificial

---

[1] This paper is partially based on the Ph.D. dissertation of the second author. Preliminary versions have been presented at the IJCAI'93 Workshop on Knowledge-Based Scheduling, Chambéry, France, September 1993, and at the EURO XIII / OR 36 conference, Glasgow, UK, July, 1994.

Intelligence (Montanari, 1974; Van Hentenryck, 1990). This approach is in some sense more general than the three others since i) it is based on the systematic use of constraints, ii) it can implement heuristic knowledge such as priority rules that can guide the search inside the feasibility domain (e.g., Bensana et al., 1988); moreover, due to its generality, the constraint satisfaction approach can be viewed as a general framework for stating and solving combinatorial optimization problems.

Developing a good predictive schedule that satisfies temporal, technological and other types of constraints is basically a search problem, the solution of which requires both powerful search heuristics and adequate means of representation (e.g., Fox and Smith, 1984; Lepape, 1985; Peng and Smith, 1986; Bensana et al., 1988; Sadeh, 1991). However, when scheduling over long horizons, considering temporal constraints, such as job release date and due date, as compulsory, may lead to rejecting an efficient schedule even when the violation of these constraints is insignificant with regard to the precision of the realistic limits of predictability. A large computation effort may be actually saved by avoiding failures with problems whose lack of feasibility is due to insignificant constraint violations.

In practical problems, constraints often prove to be more or less relaxable or are subject to preferences; this is typically true for due-date constraints in scheduling (e.g., in Fox(1987) or Sadeh (1991)). Fuzzy sets appear as a suitable framework for the representation of such flexible temporal constraints. Besides, some scheduling parameters like the durations of the tasks may be ill-known, because of the uncertainty pervading the process, and can be represented by possibility distributions. This paper presents a constraint-guided approach to job-shop scheduling based on possibility theory (Zadeh, 1978; Dubois and Prade, 1988) which can be understood as a fuzzy extension of the one formerly proposed by Erschler et al. (1976) and more recently developed, in the setting of Artificial Intelligence, by Erschler and Esquirol (1986), and Erschler et al. (1989, 1991). The approach advocated in this series of papers emphasizes the use of constraint propagation rules that may enforce sequencing decisions among tasks sharing common resources. In the present paper, flexible temporal constraints over release dates, due dates and durations, as well as uncertain durations are expressed and handled in the framework of possibility theory. This approach is a direct application of a general approach to flexible Constraints Satisfaction Problems (FCSPs) as described in Dubois et al. (1993,1994). The merits of exploiting flexibility in constraint-directed scheduling are twofold (e.g., Dubois, 1989): avoiding the arbitrary selection of a solution when the constraints are loose (as a classical constrained-directed approach would do) and avoiding infeasibility due to tight constraints when a slight relaxation of these constraints would produce a worthwhile solution. These advantages are expected to go along with some computational savings due to enhanced capabilities of guiding the search process towards interesting solutions. The aim of this paper is to rigorously formulate softly constrained jobshop scheduling problems with possibly uncertain duration of tasks.

The next section presents how to formulate constrained scheduling problems, in the classical CSP framework. Section 3 formulates a softly constrained non-preemptive, non-cumulative jobshop scheduling problem with known durations of operations. It takes into account flexible temporal constraints (release and due dates). In section 4, this formulation is extended to flexible durations (which are under control), as well as uncertain durations (when some parameters are beyond control), whose value can only be fuzzily estimated. It is based on the idea that possibility theory can model preference as well as uncertainty. The remainder of the section then explains how fuzzy non conjunctive graphs of linear inequalities can deal with the representation of such flexible constraints as well as uncertainty using a unique formulation. In Section 5, a solving scheme is presented, which relies on three basic procedures: consistency enforcing, tree search and look-ahead analysis. These procedures are fuzzy extensions of the ones used in constraint-directed search methods.

2. Scheduling as a Constraint Satisfaction Problem

A typical scheduling problem can be described as follows: a set J of jobs must be performed by means of a set of resources. Each job j requires the scheduling of a set Ω_j of operations according to a process plan that specifies a partial ordering among these operations (precedence constraints). Let Ω denote the set of operations. Once started, operations cannot be interrupted. In the simplest situation, each operation O_i must be performed by a given resource and has a precise duration t_i. Each resource can only process one operation at a time: capacity constraints between two operations requiring the same resource express that these two operations cannot overlap in time. Let s_i denote the starting time of O_i, that the scheduling procedure must compute. Propagating the release dates and the due dates of job j over operations in Ω_j yields time windows $[r_i,d_i]$ where each operation O_i must take place; r_i is its release date (earliest starting time) and d_i its due date (latest ending time). The different constraints that bear on the starting times of each operation translate into linear inequalities of the type:

precedence constraints $P_{i\to k}$: $s_k - s_i \geq t_i$ (O_i before O_k)
capacity constraints $C_{i\leftrightarrow k}$: $s_k - s_i \geq t_i$ or $s_i - s_k \geq t_k$ (O_i and O_k cannot overlap).
release date constraints R_i: $s_i \geq r_i$
due date constraints D_i: $s_i + t_i \leq d_i$.

Note that the capacity constraints are not linear stricto sensu due to the presence of a disjunction in capacity constraints. Non-disjunctive constraints form the conjunctive part of the problem. Of course this is not the most general form of scheduling problem, but this one is quite often found in the literature, and is known to be very combinatorial.

As pointed out in (Sadeh, 1991), scheduling problems can be understood as particular Constraint Satisfaction Problems (CSP's) (Mackworth, 1977; Montanari, 1974). A constraint satisfaction problem is defined by means of a set V of decision variables $v_1, v_2,\ldots, v_n$, each with a domain A_i, for i = 1, n; and a set of constraints $C_1,C_2,\ldots,C_m$. Each constraint C_j involves a subset V_j of variables and is modelled as a relation R_j, that is a subset of admissible tuples $(a_{j1},a_{j2},\ldots,a_{jk})$ in the Cartesian product $A_{j1} \times A_{j2} \times \ldots \times A_{jk}$ of domains of the k variables in V_j. The problem is then to find a feasible tuple $(a_1,a_2,\ldots,a_n)$ which satisfies all the constraints. This formulation is very general since nothing is assumed about the form of the constraints. Artificial Intelligence has devised general languages and procedures for stating and solving CSP's especially in the case of finite domains, including generic constraint propagation algorithms that make the search more efficient (e.g.Van Hentenryck, 1989). Many results exist on so-called constraint networks (where constraints are binary, i.e.involve only two variables at a time), and, in the infinite domain case when the variable domains are intervals (Davis, 1987) In this section, the constraint analysis approach to scheduling problems is related to the current CSP technology.

Dechter, Meiri and Pearl (1991) define Temporal Constraint Satisfaction Problems (TCSPs) as binary Constraint Satisfaction Problems (CSPs) whose variables take real values. In TCSPs, every constraint T_{ij} relating a pair of variables (x_i,x_j) is defined by a set of intervals $\{I_{ij}^1, \ldots, I_{ij}^n\}$ meaning that $x_j - x_i$ must belong to $I_{ij}^1 \cup \ldots \cup I_{ij}^n$. Similarly, the domain of each variable x_i is a unary constraint T_{ii} defined by a set of intervals. The conjunctive part of a TCSP (defined by the constraints involving at most one interval) is a Simple Temporal Problem (STP). Usual CSP's constraint propagation algorithms (like AC-3, PC2) can be applied over a TCSP.

A scheduling problem can be easily represented by a TCSP whose variables are the starting times of the tasks (adding a dummy variable, s_0 which stands for the beginning of the schedule). The constraints pertaining to the problem can be described as follows:

– release and due date constraints relate each s_i to s_0: $s_i - s_0 \in [r_i, d_i - t_i]$
– precedence constraints $P_{i\to k}$: $s_k - s_i \in [t_i,+\infty)$
– capacity constraints $C_{i\leftrightarrow k}$: $s_k - s_i \in [t_i,+\infty) \cup (-\infty,-t_k]$

Considering only the conjunctive part of the graph, each precedence constraint $P_{i\to k}$ implies that the temporal window associated to O_k (resp. O_i) must be such that $r_k \geq r_i + t_i$ (resp. $d_i \leq d_k - t_k$). Hence, the temporal windows can be calculated as follows:

$$r_k := \max\{r_i + t_i \text{ , for all i such that } P_{i\to k}\} \qquad (1)$$
$$d_i := \min\{d_k - t_k \text{ , for all k such that } P_{i\to k}\} \qquad (2)$$

This algorithm is the classical shortest or longest path algorithms of deterministic PERT-like networks. It takes advantage of the acyclicity of the graph to produce an efficient ordering for calculating the temporal windows. It is well known to be polynomial in complexity. The r_k's are updated along precedence constraints, and the d_i's are updated backwards. This method guarantees that the best among the earliest (resp. latest) starting times according to the precedence constraints can then be obtained when assigning to each s_i the lowest (resp. greatest) date among its best possible values. In the context of constraint propagation, the above calculation is called a consistency-enforcing procedure and more precisely an *arc-consistency* procedure. Indeed, enforcing arc-consistency in a TCSP consists of iteratively considering each pair of variables (x_i, x_j) related by a constraint T_{ij} and applying the updating pattern $T_{ii} := T_{ii} \cap ((T_{jj} \ominus I_{ij}^1) \cup \ldots \cup$

$(T_{jj} \ominus I_{ij}^{n})$) and $T_{jj} := T_{jj} \cap ((T_{ii} \oplus I_{ij}^{1}) \cup \ldots \cup (T_{ii} \oplus I_{ji}^{n}))$, where $\oplus$ and $\ominus$ denote the interval addition and subtraction, respectively. If T_{ij} represents a precedence constraint, these updating patterns respectively correspond to backward and forward propagation of the starting times as in (1-2): they ensure that the set of possible starting times of each operation is consistent with the possible starting times of each of its neighbours. (1-2) is also an improved arc-consistency procedure, limited to the conjunctive part of the TCSP since the update is achieved in one step.

The difficulty of the jobshop scheduling problem lies in the necessity of "breaking" the disjunctive constraints so as to get a purely conjunctive problem that is easily solved via (1-2).It means turning all capacity constraints $C_{i\leftrightarrow k}$ into precedence constraints $P_{k\rightarrow i}$ or $P_{i\rightarrow k}$. It comes down to finding a sequence of operations to be performed on each machine. This sequence must be feasible in the sense that the associated conjunctive problem must have a solution in terms of starting times. Finding the sequences of operations on machines is a problem dual to the one of finding starting times for operations. It is a difficult combinatorial problem on a finite domain, while finding the starting times is an easy STP (albeit on an infinite domain) once the sequences of operations are found. In order to find such a feasible solution, an obvious method is a depth-first exploration method with backtrack. It will be computationally inefficient. To make it more efficient one must use look-ahead procedures that quickly check some consequences of selecting a new precedence constraint. Such dedicated procedures have been proposed in the framework of the constraint analysis approach to jobshop scheduling (Erschler et al., 1976; Erschler and Esquirol, 1986). For each disjunction $C_{i\leftrightarrow k}$ also called a conflict (O_i precedes O_k OR O_k precedes O_i), a test is performed as follows:

Test-1(i,k) If $d_k - r_i < (t_i + t_k)$ then O_i cannot precede O_k

Clearly such a test can lay bare a precedence constraint which, if violated leads to a solution which is not feasible (enforcing that O_i must precede O_k when $d_k - r_i < (t_i + t_k)$ prevents operations from being performed in the time window $[r_i, d_k]$). If both conditions of Test-1(i,k) and Test-1(k,i) are verified, this is a contradiction and the scheduling problem itself is not feasible. If none of these tests is positive, then the conflict remains unsolved (since nothing can be inferred on the precedence between O_i and O_k in this case). This kind of test provides only a necessary condition of feasibility. For instance, consider a case involving three tasks conflicting for the same resource: $t_i = t_k = t_x = 2$, $r_k = 0$, $r_i = r_x = 1$ and $d_i = d_k = d_x = 6$. It is obtained that Test-1(i,k) and Test-1(k,i) are both negative, although no scheduling placing O_i before O_k is feasible. More elaborate tests of the conflicts can be performed, taking more than two operations into account. Constraint analysis rules like those proposed in (Erschler and Esquirol, 1986; Erschler et al., 1989) are of that kind. The difficulty of using these more refined tests is that they may generate new capacity constraints involving more than two tasks (e.g., "O_i must precede O_k or O_x"), i.e., general disjonctive constraints. These new obtained constraints are not easy to use in a look-ahead strategy.

A new kind of test is proposed here, involving more than two operations and which do not present this drawback. For instance, consider three operations O_i, O_k and O_x, and the following:

Test-2(i,k): if either $d_k - r_i < (t_i + t_k)$

or $\max(d_k - r_i, d_k - r_x, d_x - r_i) < t_i + t_k + t_x$

then O_i cannot precede O_k

where the three terms in the maximum operation pertain to the respective sequences: $O_i/O_x/O_k$, $O_x/O_i/O_k$ and $O_i/O_k/O_x$. Coming back to the above numerical example, Test-2(i,k) is positive, which enforces the other precedence constraint, i.e., that O_k must precede O_i. Clearly this test is more time-consuming than Test-1(i,k) but is likely to cause more early backtracks. This test can be extended to more than one varying operation O_x.

The above tests correspond to so-called path-consistency in CSP's. Enforcing path-consistency in a TCSP consists of iteratively considering each 3-tuple of variables (x_i, x_j, x_k) related by the constraints T_{ik} and T_{kj} and applying the updating pattern :

$$T_{ij} := T_{ij} \cap (\bigcup_{a,b} I_{ik}{}^{a} \oplus I_{kj}{}^{b}).$$

When $T_{ij} = C_{i\leftrightarrow j}$ and x_k stands for the starting time of an operation O_k, the updating pattern ensures that the constraint between O_i and O_j is coherent with O_k: it corresponds to the second look ahead test Test-2(i,k). When $x_k = s_0$, and $T_{ik} = C_{i\leftrightarrow k}$ this updating pattern corresponds to the first constraint analysis test Test-1(i,k). Indeed, $I_{i0} = [-d_i + t_i, -r_i]$, $I_{0k} = [r_k, d_k - t_k]$ and:

$$[-d_i + t_i, -r_i] \oplus [r_k, d_k - t_k] = [r_k - d_i + t_i, d_k - t_k - r_i].$$

Moreover:

$$\begin{aligned}
&([t_i,+\infty) \cup (-\infty,-t_k]) \cap [r_k - d_i + t_i, d_k - t_k - r_i] \\
&\quad = [\max(t_i, r_k - d_i + t_i), d_k - t_k - r_i] \text{ (decision } O_i \text{ before } O_k) \\
&\qquad\qquad \text{when } t_k + t_i \le d_k - r_i \text{ and } t_k + t_i > d_i - r_k \\
&\quad = [r_k - d_i + t_i, \min(-t_k, d_k - t_k - r_i)] \text{ (decision } O_k \text{ before } O_i) \\
&\qquad\qquad \text{when } t_k + t_i > d_k - r_i \text{ and } t_k + t_i \le d_i - r_k
\end{aligned}$$

The main difference between TCSP's updating pattern and the look-ahead tests is that the TCSP path-consistency algorithm modifies the set of intervals attached to a constraint even when no decision can be taken (hence, a capacity constraint originally defined by two disjoint intervals can be transformed into a constraint involving more intervals). This modification is then propagated to other 3-tuples of tasks. Constraint analysis can only reduce the number of intervals attached to a constraint, making nothing when it should be increased. In other terms, constraint analysis in scheduling problems is a weak version of path-consistency in TCSPs, also called 3-Boundary consistency by Lhomme(1993).

Finally, the standard solving method (backtrack search) is the same in TCSPs and in scheduling problems: it consists in considering the dual problem in order to choose a decision for each capacity constraint (i.e., to choose an interval for each temporal constraint T_{ik}). Once a decision has been taken, it is propagated (i.e., arc-consistency enforcing and constraint analysis procedures are applied) as it is the case in classical CSP when a look-ahead procedure is used.

Hence, the above type of scheduling problems are particular TCSPs, which are themselves particular CSPs. Moreover, solving procedures for scheduling problems can be understood as specialized (and thus improved) versions of some classical CSP algorithms. This remark suggests adapting the results obtained in the CSP domain in order to improve the resolution of scheduling problems — being aware of the specificity of such problems, especially when designing search heuristics: Sadeh (1991) has shown that translating without modification CSP heuristics to the scheduling domain is inefficient. A more promising research area for both domains is the study of flexible problems.

3. The fuzzily constrained scheduling problem

The idea of fuzzy constraint is not new. It goes back to the seminal paper of Bellman and Zadeh (1970). In this paper the suggestion that a fuzzy set may be a natural model of a soft constraint is put forward. It is also suggested that an optimal solution to a fuzzily constrained problem is a solution such that its membership grade to the intersection of the fuzzy sets modelling the constraints is maximal. It comes down to maximizing the degree of satisfaction of the least satisfied constraint. This strategy is quite in accordance with the notion of a constraint whose violation cannot be counterbalanced by the satisfaction of other constraints. However, since 1970, the fuzzy constraint idea has been mainly used in stating soft versions of linear programming (Zimmermann 1976; Slowinski and Teghem, 1990; Sakawa, 1993). Very few authors have considered fuzzy formulations of jobshop scheduling problems, even though fuzzy PERT has been studied quite early, and the third author made some early related contributions (Prade, 1979). The use of possibility theory in jobshop scheduling has been further discussed by Kerr and Walker(1989) and (Dubois, 1989); the latter contains preliminary results that are elaborated upon in the present paper. More recently, simple forms of fuzzy flow-shop scheduling problems have been solved by Ishii et al.(1992) and Ishibushi et al.(1994). Besides, the link between the ideas of Bellman and Zadeh(1970) and the mainstream literature on constraint satisfaction problems has been made by Freuder and Snow (1990), but fuzzy sets are not yet widely used in CSP's to-date (see Dubois et al 1993, 1994, for a bibliography). Fuzzy scheduling problems as described in this paper can be viewed as dedicated Fuzzy Constraint Satisfaction Problems.

3.1. Flexible temporal constraints

Release and due dates of jobs are often subject to preference. For instance job j must absolutely be completed at the latest completion date d^{sup}_j (e.g., the date after which the customer cancels his order). Moreover it should preferably be completed before the due date d^{inf}_j, or as soon as possible after this due date. Similarly, it is better to start job j after its preferred release date r^{sup}_j, because the corresponding raw material will be available then, while it is impossible to start it before the earliest acceptable release date r^{inf}_j because the raw material will not be ready ($r^{inf}_j \le r^{sup}_j < d^{inf}_j \le d^{sup}_j$). The requirement about the release date (resp. due date) associated to job j is no longer crisp but can be modeled by means of a fuzzy number R(j) (resp. D(j)) as in Figure 1. Namely the membership function $\mu_{R(j)}$ (resp. $\mu_{D(j)}$) is such that $\mu_{R(j)}(r^{sup}_j) = 1$, and $\mu_{R(j)}(r^{inf}_j) = 0$, and increasing when r_j goes from r^{inf}_j to r^{sup}_j (resp. $\mu_{D(j)}(d^{inf}_j) = 1$, and $\mu_{D(j)}(d^{sup}_j) = 0$, and decreasing when d_j goes from d^{inf}_j to d^{sup}_j). Hence, the temporal window in which the job must take place is the fuzzy interval [R(j),D(j)] as in Figure 1.

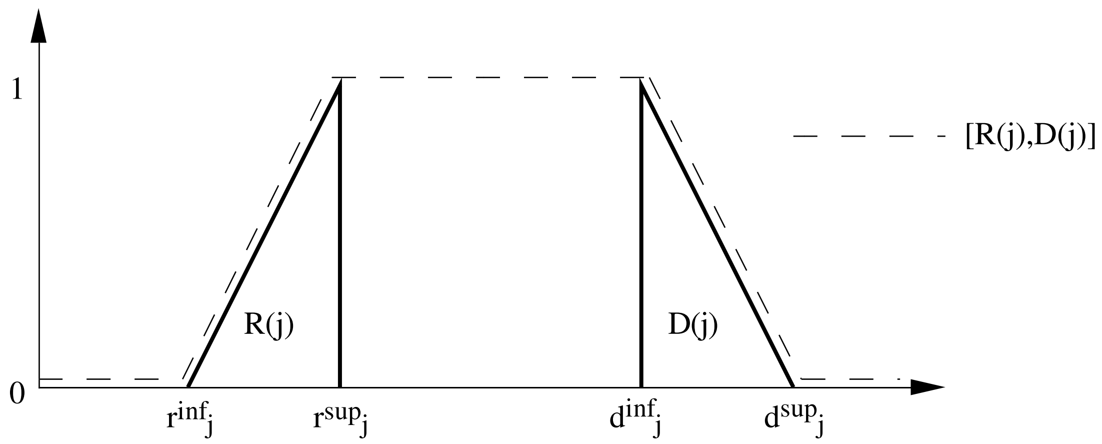

Figure 1: Fuzzy time horizon for job j

In order to model such fuzzily bounded intervals, let us first recall some results from possibility theory (Dubois and Prade, 1988). Consider a parameter x whose values are restricted by a fuzzy set A, so that its possibility distribution π_x is taken as equal to the membership function μ_A.The possibility of the event "$x \in P$" denoted $\Pi(x \in P)$ is the degree of intersection between A and P, where fuzzy set intersection is defined by the minimum:

$$\Pi(x \in P) = \sup_u \min (\mu_A(u), \mu_P(u)). \qquad (3)$$

It estimates to what extent "$x \in P$" is possibly true, or, equivalently, to what extent "$x \in P$" is consistent with the information "$x \in A$" modelled by $\pi_x = \mu_A$. Note that in (3) P can be a fuzzy set as well.

The dual measure of necessity of "$x \in P$" denoted $N(x \in P)$ evaluates to what extent A is included in the core of P, that is the set $c(P) = \{u , \mu_P(u) = 1\}$ or, in other terms, to what extent "$x \in P$" is certainly true, i.e., is entailed by "$x \in A$":

$$N(x \in P) = \inf_u \max (1 - \mu_A(u), \mu_P(u)) = 1 - \Pi(x \in \bar{P}). \quad (4)$$

where $\bar{P}$ denotes the fuzzy complement of P ($\mu_P = 1 - \mu_{\bar{P}}$). Indeed, $N(x \in P) = 1$ if and only if the support of A, namely the subset $\{u , \mu_A(u) > 0\}$, is included in c(P), i.e., if all the more or less possible values of x are among the values that totally satisfy P. In particular, if x is a real variable, A a fuzzy interval and p a crisp number, it holds that (Dubois and Prade, 1988):

$$\Pi(x \geq p) = \Pi(x \in [p,+\infty)) = \sup_{u \geq p} \mu_A(u) = \mu_{(-\infty,A]}(p)$$
$$N(x \geq p) = N(x \in [p,+\infty)) = \inf_{u<p} 1 - \mu_A(u) = \mu_{(-\infty,A[}(p)$$
$$\Pi(x \leq p) = \Pi(x \in (-\infty,p]) = \sup_{u \leq p} \mu_A(u) = \mu_{[A,+\infty)}(p)$$
$$N(x \leq p) = N(x \in (-\infty,p]) = \inf_{u>p} 1 - \mu_A(u) = \mu_{]A,+\infty)}(p)$$

where $(-\infty,A]$, $(-\infty,A[$, $[A,+\infty)$, $]A,+\infty)$ respectively denote the set of points possibly before A, necessarily before A, possibly after and necessarily after A (see Figure 2).

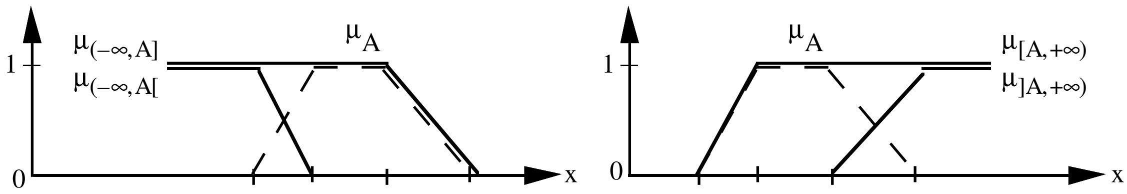

Figure 2: (a) points possibly/necessarily before A; (b) points possibly/necessarily after A

The fuzzy temporal window [R(j),D(j)] is defined as $(-\infty, D(j)] \cap [R(j), +\infty)$. When the requirement about the release date r_j of job j is fuzzy, the constraint $s_j \geq r_j$ becomes flexible and its satisfaction by a choice of the starting time s_j can be a matter of degree. Indeed, the coefficient:

$$\Pi(s_j \geq r_j) = \mu_{[R(j),+\infty)}(s_j) \tag{5}$$

can be understood as the satisfaction degree of the release date constraint for job j starting at time s_j. Similarly, when the requirement about the due-date of job j is fuzzy, the constraint $e_j \leq d_j$ on the ending time e_j of the job is satisfied to degree:

$$\Pi(e_j \leq d_j) = \mu_{(-\infty,D(j)]}(e_j) \tag{6}$$

$\Pi(e_j \leq d_j)$ indicates to what extent there exists an acceptable value for d_j greater than e_j, given that d_j is restricted by D(j): it is equal to 1 if $d^{inf}_j \geq e_j$, that is to say if the job is completed before the preferred due date. If the job finishes after the latest acceptable completion date ($d^{sup}_j \leq e_j$), then $\mu_{(-\infty,D(j)]}(e_j) = 0$. Otherwise, the closer e_j to the preferred due date, the higher $\mu_{(-\infty,D(j)]}(e_j)$. In other terms, the due date constraint is a flexible constraint and the fuzzy set models how the due date can be relaxed from the customer's preferred due date to the latest acceptable completion date. Note that in this example, fuzzy sets D(j) and R(j) model preference, not uncertainty, in the sense that D(j) and R(j) describe the wishes of the decision-maker regarding the schedule, not his guess pertaining to the actual starting time and ending time of job j. This interpretation of fuzzy numbers in terms of preference profiles, which also applies to durations of tunable operations as in the next section, has been put forward by Dubois (1987), but also Wood et al. (1992) in the domain of mechanical engineering.

At this point it is useful to recall the calculus of fuzzy intervals. Let A and B be two fuzzy intervals. The sum $A \oplus B$ and the difference $A \ominus B$ of two fuzzy intervals is defined by (e.g., Dubois and Prade, 1980; 1988)

$$\begin{aligned}\mu_{A\oplus B}(z) &= \sup_{x,y:z=x+y} \min(\mu_A(x), \mu_B(y)) \\ &= \sup_x \min(\mu_A(x), \mu_B(z-x))\end{aligned} \tag{7}$$

$$\begin{aligned}\mu_{A\ominus B}(z) &= \sup_{x,y:z=x-y} \min(\mu_A(x), \mu_B(y)) \\ &= \sup_y \min(\mu_A(z+y), \mu_B(y)).\end{aligned} \tag{8}$$

When B = n (precise value), $\mu_{A\ominus n}(z) = \mu_A(z + n)$, noticing that $(-\infty,A] \ominus n = (-\infty, A \ominus n]$.

Sup-min convolutions (7) and (8) have different meanings according to whether the fuzzy variables underlying A and B are controllable or not. If controllable, the membership functions represent preference profiles and $\mu_{A\oplus B}(z)$ represents the optimal preference level that can be attained for the assignment of values to x and y respecting the preference profiles A and B and the constraint z = x + y; this optimum is of the egalitarist type since it is such $\mu_A(x) = \mu_B(y) = \mu_{A\oplus B}(z)$. On the contrary if A and B represent more or less plausible values of ill-known parameters that cannot be tuned, $\mu_{A\oplus B}(z)$ is the level of plausibility that the sum x + y will take value z given the *plausibility* profiles A and B for x and y. This calculation presupposes that the possible links between parameters x and y are unknown, so that assuming a value for x should not a priori induce any supplementary restriction on the possible value of y, and conversely. The two parameters are then called *non-interactive* (Zadeh, 1975). The calculation proceeds by fixing a plausibility level α and performing interval (best- and worst case) analysis on the level cuts $A_\alpha = \{x, \mu_A(x) \geq \alpha\}$ and $B_\alpha = \{y, \mu_B(y) \geq \alpha\}$. $\mu_{A\oplus B}$ is then obtained by moving the threshold in the unit interval.

3.2. Feasible schedules

A solution to a crisp scheduling problem with precise durations is typically an assignment $(s_1, \ldots, s_n)$ of starting times of all the operations. It must satisfy precedence constraints, capacity constraints, release and due date constraints. While release and due dates are flexible, capacity and precedence constraints remain crisp. An assignment satisfying precedence and capacity constraints satisfies the fuzzy scheduling problem insofar as it satisfies the least satisfied temporal constraint. The global satisfaction level depends on the chosen starting times for operations. It is defined as:

$$\begin{aligned}\text{Sat} &= 0 \text{ if a capacity or a precedence constraint is violated in the processing} \\ &= \min_{j\in J} (\min \mu_{[R(j),+\infty)}(s_j), \mu_{(-\infty,D(j)]}(e_j)) \text{ otherwise}\end{aligned} \tag{9}$$

where s_j and e_j are respectively the starting and the ending dates of job j. These ending dates depend in a non-trivial way on the starting times of the jobs, on the durations of elementary operations, on the chosen sequences of operations on the various machines and on the chosen starting times of operations. *Sat* represents the minimal fraction of the flexibility ranges $d^{sup}_j - d^{inf}_j$ (resp. $r^{sup}_j - r^{inf}_j$) which are left between the completion times e_j (resp. starting times s_j) of the jobs and their latest acceptable completion times (resp. earliest release dates). If all the flexibility ranges are taken equal, the best schedules are those that minimize the tardiness and earliness of all the jobs. In other terms, the present approach is looking for a temporally safe schedule. The degrees of membership are not interpreted in terms of cost (contrary to Sadeh, 1991). In scheduling problem approaches like those defined in (Fox and Smith, 1984; Fox 1987, Sadeh, 1991), a high degree of satisfaction of a constraint (e.g., total satisfaction of due date for a job) can counterbalance a low degree of satisfaction of another constraint (e.g., almost violation of latest acceptable completion date for another job). Here, satisfaction degrees cannot be interpreted in terms of costs, but in terms of safety ranges.

In order to lay bare the role of the sequences of operations on machines, fuzzy temporal windows for the operations O_i can be calculated from the knowledge of R(j) and D(j) for $j \in J$ and the current precedence constraints between operations. The R(i)'s and D(i)'s have to be computed via an extension of (1-2) to fuzzy numbers. Considering only the conjunctive part of the graph, each precedence constraint $P_{i \to k}$ implies that the fuzzy temporal window associated to O_k (resp. O_i) must be such that $r_k \geq r_i + t_i$ (resp. $d_i \leq d_k - t_k$). Hence, the temporal windows can be computed as follows (Dubois and Prade, 1988):

$$R(i) := \widetilde{\max}\{ R(k) \oplus t_k, \text{ for all k such that } P_{k \to i} \} \tag{10}$$

$$D(i) := \widetilde{\min}\{ D(k) \ominus t_k, \text{ for all i such that } P_{i \to k} \} \tag{11}$$

where $\widetilde{\max}$ and $\widetilde{\min}$ are versions of the maximum and minimum operations extended to fuzzy arguments, i.e., replacing addition by max and min respectively in (7). A linear algorithm has been implemented which computes the fuzzy temporal windows according to the precedence constraints. This algorithm is an adaptation of classical shortest or longest path algorithms to fuzzy PERT-like networks. These algorithms can accommodate the case when the durations t_k are replaced by fuzzy numbers T(k) in (10-11). Similar methodologies for propagating fuzzy upper and lower bounds of starting times in activity networks have been proposed by Chanas and Kamburovski (1981), Gazdik (1983) and more recently Nasution (1993). Lootsma (1989) has made a critical comparison between fuzzy and stochastic PERT networks, where he points out the fact that the fuzzy approach ignores the dependencies induced by the topology of the network representing precedence constraints. In fact, the criticism from probability theory towards fuzzy arithmetic makes some sense when the durations, release dates and due-dates are considered as uncontrolable, because when due to randomness, this randomness is more naturally modeled via probability theory. In particular, probabilistic methods easily account for random compensation phenomena, assuming independent variables. Clearly, the max-min fuzzy arithmetic only refines the crude worst- and best-case interval analysis, by means of various levels of plausibility. Dependencies between fuzzy variables may be captured by changing the minimum into another fuzzy conjunction, such as product, for instance. However when the lack of knowledge of precise durations and the like accounts for suspended decisions, then the membership functions represent preference profiles and fuzzy arithmetics is fully justified as implementing fuzzy constraint propagation (in the sense of Dubois, Fargier and Prade (1994)); but probability theory does not apply at all in such a situation.

Note that here, the two computations pertaining to earliest starting times R(k) and latest starting times D(i) are rather independent because release dates and due-dates of jobs are separately specified. This paper is not interested in the critical path method where the due-date of each operation is obtained after a forward propagation in the precedence graph. In the latter case the update of the D(i)'s takes place after the calculations of the R(k)'s, and in this second step the involved fuzzy variables are related since the fuzzy ending time obtained in the first step is taken as a fuzzy due-date which then depends on the fuzzy durations of operations. Some solutions to this problem are proposed by Nasution (1993).

Suppose the operation duration is precisely known as t_i. Hence, the set of more or less admissible starting times for the operation i according to the two flexible temporal constraints that determine its time window is the fuzzy interval $[R(i),D(i) \ominus T_i]$ since $\mu_{(-\infty,D(i)]}(s_i + t_i) = \mu_{(-\infty,D(i)\ominus t_i]}(s_i)$ (see Fig. 3).

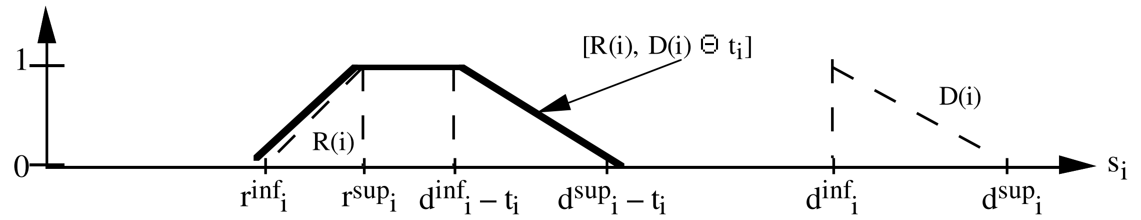

Figure 3: possible values for s_i given the temporal constraints on operation i.

Equations (10)-(11) operate a projection of the temporal constraints pertaining to jobs over the space of starting times of operations. It makes implicit constraints on starting times explicit, and (9) can be expressed as:

$$\begin{aligned} Sat(s_1, \ldots, s_n) &= 0 \text{ if } (s_1, \ldots, s_n) \text{ violates a precedence or a capacity constraint} \\ &= \min(\min_{i=1,n} \mu_{[R(i),+\infty)}(s_i), \min_{i=1,n} \mu_{(-\infty,D(i)]}(s_i + t_i)) \text{ otherwise.} \end{aligned} \quad (12)$$

In the fuzzily constrained job-shop scheduling problem, solutions are not equally preferred: satisfaction degrees induce a total ordering over the solutions of the problem defined by capacity and precedence constraints. A fuzzy job-shop scheduling problem is in fact a constrained optimization problem for which the best solutions are those requesting the least relaxation of release dates or due dates. In any case the solutions violating any earliest release date or latest completion date are not acceptable (sat = 0) whereas the solutions satisfying preferred release and due dates (if they exist) are the best (sat = 1). Otherwise, an implicit relaxation of flexible constraints is performed, achieving a trade-off between antagonistic constraints in the spirit of (Descottes and Latombe, 1985): this framework allows for the treatment of partially inconsistent problems. In fact, the satisfaction degree of the best solution evaluates to what extent there is an solution satisfying all the constraints. The feasibility degree of the problem can be defined by:

$$\mathrm{Cons} = \max_{(s_1,\ldots,s_n)} \mathrm{Sat}(s_1, \ldots, s_n) \tag{13}$$

When the constraints are partially inconsistent, $0 < \mathrm{Cons} < 1$, since no assignment can perfectly satisfy all the constraints. The lack of compensation between levels of satisfaction of constraints is due to the nature of the problem. Namely one takes it for granted that no constraint should be completely violated by solutions that are accepted as not completely unfeasible. For instance any additive aggregation of membership grades in (9) might result in an optimal solution that completely violates some constraint for the sake of fully respecting others. Hence the difference between the satisfaction profile of a constraint and an objective function in the usual sense lies in the absence of trade-off when performing aggregation of partial satisfaction levels, as opposed to usual practice in multiple criteria optimization. Clearly any fuzzy conjunction operation $*$ such that $a * b \leq \min(a,b)$ can be used (e.g., the product of membership grades). However these fuzzy conjunctions are not idempotent, hence would not tolerate dependent constraints: writing the same constraint twice makes a difference, for instance. In the present problem, the degrees of membership of starting times of operations to the fuzzy intervals induced by fuzzy temporal constraints are not independent since time-windows on operations are induced by more global temporal constraints on jobs. Hence the use of the minimum operation looks compulsory in (12) because it copes with redundancy. If min were changed into product, problems (9) and (12) would not be equivalent .

4. Fuzzy durations

The above framework can be extended to problems with operations involving fuzzy durations, depending on the meaning conveyed by the fuzziness. First, durations can be subject to preferences, when they are decision variables under control, just as starting times. Second, durations can also be partially out of control; this case will be considered in sections 4.2 to 4.4.

4.1 Controlable durations

Controlable durations, are subject to preferences, and may indeed be determined by tuning the machine on which the operation is performed (for instance, tuning the speed of a machine-tool affects the machining time). For a given operation to be optimally performed, ideal values of the tuning parameters exist. More generally, fuzzy ranges that constrain feasible parameter values, with various preference levels, can be modelled by fuzzy numbers (see Dubois, 1987). Two conflicting requirements for the tuning can be envisaged: the shorter the duration, the better for the sake of meeting scheduling constraints; however, the optimal tuning parameter values may lead to a longer processing time ensuring a better quality of the processing.

The possible durations of operation O_i may be described by a minimal duration $t^{inf}{}_i$ and a preferred duration $t^{sup}{}_i$. A fuzzy number T(i) can be defined, like R(i) and D(i). The duration t_i of each operation O_i is then a decision variable whose allowed values are restricted by the flexible constraint $t_i \in T(i)$, that must be taken into account when computing the satisfaction degree of an assignment $(s_1, \ldots, s_n, t_1, \ldots, t_n)$:

$$\begin{aligned} \mathrm{Sat}\,(s_1, \ldots, s_n, t_1, \ldots, t_n) &= 0 \text{ if a capacity or a precedence constraint is violated.} \\ &= \min_{i=1,n} \min(\mu_{T(i)}(t_i), \mu_{[R(i),+\infty)}(s_i), \mu_{(-\infty,D(i)]}(s_i + t_i)) \text{ otherwise.} \end{aligned} \tag{14}$$

This approach is different from the usual cost-project minimisation (CPM) approach (e.g. Bellman et al., 1982) where durations are tuned so as to minimise the sum of the cost for completing the operations. Here, the membership grades of durations reflect quality of processing and do not compensate.The formulation can be simplified in such a way that it is possible to select starting times first, and then to choose the optimal durations in accordance with these starting times. Assume that $(s_1, \ldots, s_n)$ is fixed and that $s_i \le s_k$ whenever a precedence constraint $P_{i\to k}$ exists. Let $\mathcal{S}_i$ be the set of operations that must start after operation O_i, that is $\mathcal{S}_i = \{O_k, P_{i\to k}\} \cup \{O_k, C_{i\leftrightarrow k} \text{ and } s_i \le s_k\}$. The value of t_i must be chosen such that $s_i + t_i \le \min_{k\in \mathcal{S}_i} s_k$. One is then led to compute the degree of membership of $\min_{k\in \mathcal{S}_i} s_k - s_i$ to the fuzzy set $[T_i,+\infty)$, i.e., for each operation i, noticing that $\mu_{[T(i),+\infty)}$ is increasing in the wide sense:

$$\mu_{[T(i),+\infty)} (\min_{k\in \mathcal{S}_i} s_k - s_i) = \min_{k\in \mathcal{S}_i} \mu_{[T(i),+\infty)}(s_k - s_i). \tag{15}$$

Similarly, the degree to which operation O_i fits in its fuzzy time window is at best

$$\sup_{t_i} \min(\mu_{R(i)}(s_i), \mu_{D(i)}(s_i + t_i), \mu_{T(i)}(t_i)) = \min(\mu_{R(i)}(s_i), \mu_{D(i)\ominus T(i)}(s_i)) \tag{16}$$

using equation (8) that defines the fuzzy subtraction. Moreover the following less obvious result can be established by projection of $\mathrm{Sat}\,(s_1, \ldots, s_n, t_1, \ldots, t_n)$ over the space of starting times (see Appendix I):

$$\mathrm{Sat}(s_1, \ldots, s_n) = \min_{i = 1,n} \min(\min_{k\in \mathcal{S}_i} \mu_{[T_i,+\infty)}(s_k - s_i), \mu_{R(i)}(s_i), \mu_{D(i)\ominus T(i)}(s_i)). \tag{17}$$

In other terms, the satisfaction degree of the best duration assignment which can be obtained from $(s_1, \ldots, s_n)$ is:

$$\begin{aligned} \mathrm{Sat}(s_1, \ldots, s_n) = \min\,(\quad & \min_{i=1,n} \mu_{[R(i),+\infty)}(s_i), \\ & \min_{i=1,n} \mu_{(-\infty,D(i)\ominus T(i)]}(s_i), \\ & \min_{P_{i\to k}} \mu_{[T(i),+\infty)}(s_k - s_i), \\ & \min_{C_{i\leftrightarrow k}} \max(\mu_{[T(i),+\infty)} (s_k - s_i), \mu_{[T(k),+\infty)}(s_i - s_k))). \end{aligned} \tag{18}$$

In equation (18) the feasibility degree of an assignment $(s_1, \ldots, s_n)$ can actually be read in terms of possibility degrees describing preference, considering that r_i (resp. d_i, t_i) is fuzzily restricted by R(i) (resp. D(i), T(i)), namely:

$$\mathrm{Sat}(s_1, \ldots, s_n) = \min\,(\quad \min_{i=1,n} \quad \Pi(r_i \in (-\infty,s_i]), \tag{19}$$

$$\begin{array}{ll} \min_{i=1,n} & \Pi(d_i - t_i \in [s_i, +\infty)), \\ \min_{P_{i\to k}} & \Pi(t_i \in (-\infty, s_k - s_i]), \\ \min_{C_{i\leftrightarrow k}} & \Pi(t_i \in (-\infty, s_k - s_i] \text{ or } t_k \in (-\infty, s_i - s_k])) \end{array}$$

Hence, an assignment of the starting times such than $Sat(s_1, \ldots, s_n)$ is maximal is first searched for. If needed, the set of possible durations T'(i) for each task O_i corresponding to a prescribed starting time assignment $(s_1, \ldots, s_n)$ can be then computed. The best duration assignment which can be obtained from $(s_1, \ldots, s_n)$ assigns to each t_i the value with the highest degree in T'(i) such that:

$$T'(i) = T(i) \cap (-\infty, D(i) \ominus R(i)] \ \cap\ [0, \min_{P_{i\to k}}(s_k - s_i)] \ \cap\ [0, \min_{C_{i\leftrightarrow k}} \max(s_k - s_i, s_i - s_k)]. \qquad (20)$$

4.2. Imprecise durations

In a second interpretation, fuzzy durations can be understood as ill-known parameters, due to possible perturbations and can be again represented by possibility distributions. Indeed, possibility distributions can be viewed as modelling uncertainty as well as preference (see Dubois, Fargier and Prade (1993) for an exposition of this statement). Everything depends upon whether the variable x attached to the possibility distribution π_x is controllable or not. If x is controllable, π_x is a preference profile describing the preferred values of x. On the contrary if x is not controllable, then π_x models an agent's uncertainty about the value x will eventually take. In this section, it is assumed that the duration of some operations may be subject to hazards or is ill-known because partially unpredictable. Hence the knowledge about the durations is actually imprecise (e.g., "O_i will have a duration of approximately 5 time units"). The usual model of ill-known durations is stochastic. If statistical data is available about this duration, this model is perfectly justified. However, in many cases, statistical data is out of reach and the assumption of identically repeated operations dubious. However some information about what duration is more plausible than another is often available. This kind of qualitative information about uncertain parameters (be they random or not) can be modelled by means of possibility distributions. They can be viewed as nested confidence intervals with varying plausibility levels.

The possibility distribution π_{t_i} describing the more or less possible values for the duration t_i of a task O_i corresponds to a trapezoidal fuzzy number T(i) represented by the 4-tuple $(\underline{t_i} \le t_{*i} \le t^*_i \le \overline{t_i})$ such that $\Pi(t_i = x) = \pi_{t_i}(x) = \mu_{T(i)}(x)$, see Figure 4. In this case, one looks for as robust as possible a schedule given the imprecision over the durations and the flexibility of other temporal constraints. The most robust schedules are those such that all the constraints are satisfied, whatever the durations eventually are.

A precedence constraint $P_{i\to k}$ will be satisfied if whatever the real value of t_i, $s_k - s_i \geq t_i$, i.e., if $s_k - s_i$ is greater than all the possible values of t_i. In other terms, the satisfaction degree of the precedence constraint is the necessity of the event $t_i \leq s_k - s_i$:

$$N(t_i \in (-\infty, s_k - s_i]) = \inf_{t_i} \max(1 - \mu_{T(i)}(t_i), \mu_{(-\infty,s_k-s_i]}(t_i)) = \mu_{]T(i),+\infty)}(s_k - s_i) \quad (21)$$

where $]T(i),+\infty)$ is the set of numbers *necessarily* after T(i). Note that it involves only the decreasing part of the possibility distribution $\mu_{T(i)}$. $N(t_i \in (-\infty, s_k - s_i]) = \alpha$ means that if the actual duration of operation O_i is strictly less than $\sup T(i)_{1-\alpha} = \sup\{x, \mu_{T(i)}(x) \geq 1 - \alpha\}$ then the precedence constraint can be respected at degree α. Clearly the higher α, the safer the solution, i.e., $\alpha = 1$ means that the value of t_i can be anything in the support of T(i).

Similarly, the satisfaction degree of a capacity constraint $C_{i\leftrightarrow k}$ is $N(t_i \in (-\infty, s_k - s_i]$ or $t_k \in (-\infty, s_i - s_k])$. Since the durations are non-interactive, the joint possibility distribution of (t_i,t_k) is $\pi_{t_it_k}(x,y) = \min(\mu_{T(i)}(x),\mu_{T(k)}(y))$, and this satisfaction degree is equal to (see Dubois and Prade, 1988):

$$\begin{aligned} &N(t_i \in (-\infty, s_k - s_i] \text{ or } t_k \in (-\infty, s_i - s_k]) \\ &\qquad = \max(N(t_i \in (-\infty, s_k - s_i]), N(t_k \in (-\infty, s_i - s_k])) \end{aligned} \quad (22)$$

Let us now study the satisfaction of a due date constraint. Assume the starting time is s_i. Since the true duration of O_i is not known exactly, the possible values of its ending time $s_i + t_i$ are described by a possibility distribution $\mu_{s_i \oplus T(i)}$, $\oplus$ denoting the addition of fuzzy quantities. What is thus requested is that there is a possible value for the due date d_i, whose feasible values are restricted by D(i), greater than *all* possible values of $s_i + t_i$. The satisfaction degree of the due date constraint is the inclusion degree of the fuzzy set $s_i \oplus T(i)$ in $(-\infty,D(i)]$ (instead of an intersection degree in case of flexible durations), in other terms it is the necessity of the fuzzy event $s_i + t_i \in (-\infty,D(i)]$ (see Dubois and Prade, 1988):

$$\begin{aligned} N(s_i + t_i \in (-\infty, D(i)]) &= \inf_{t_i} \max(1 - \mu_{T(i)}(t_i), \mu_{(-\infty,D(i)\ominus s_i]}(t_i)) \\ &= N(t_i \in (-\infty, D(i) \ominus s_i]) \end{aligned} \quad (23)$$

instead of its possibility degree $\Pi(d_i - t_i \in [s_i,+\infty)) = \Pi(s_i + t_i \in (-\infty,D(i)])$ (see equation (19)).

In summary, the satisfaction degree of a solution $(s_1, \ldots, s_n)$ is defined by:

$$\begin{aligned} \mathrm{Sat}(s_1, \ldots, s_n) = \min(\ & \min_{i=1,n} \ \Pi(r_i \in (-\infty,s_i]), \\ & \min_{i=1,n} \ N(t_i \in (-\infty, D(i) \ominus s_i]), \\ & \min_{P_{i\to k}} \ N(t_i \in (-\infty, s_k - s_i]), \\ & \min_{C_{i\leftrightarrow k}} \ N(t_i \in (-\infty, s_k - s_i] \text{ or } t_k \in (-\infty, s_i - s_k])). \end{aligned} \quad (24)$$

Hence, when fuzzy durations represent imprecise knowledge about the *non controllable* duration of a task, the satisfaction degree of due dates, precedence and capacity constraints are necessity degrees which are motivated by the attempt to get a robust feasibility of the schedule, facing hazardous events. This is in total contrast with the case when t_i is a controllable decision variable: the satisfaction degrees of due dates, precedence and capacity constraints are possibility degrees because it is always possible to choose the best values of durations. It is then enough that *there exist* durations satisfying the constraints in the best possible way, hence the feasibility is expressed in terms of possibility only.

4.3 Uncertainty versus preference: an illustrative example

In order to better figure out the drastic difference between a fuzzy set modelling preference and a fuzzy set modelling uncertainty, consider a very simple example of scheduling. In this example the combinatorial difficulty is ruled out so as to focus on modelling acpects only. Suppose Tom wants to attend a meeting in the morning at 8 a.m. He wants to know when to get up, so as to arrive on time. He has to take the bus to reach the venue of the meeting, and the bus ride takes about 1 hour. Besides he does not want to leave his house too early, say before 7, and in any case not earlier than 6:30 h.. Clearly the time when he leaves his house is his choice. He can express preferences on this decision variable by means of a linear fuzzy interval R with a nondecreasing membership function, lower modal value 7 and support [6.5, 12]. As for the travel time, Tom has no control on it. It depends on the waiting time (not more than 15 mn) and the traffic situation, which never causes more than a 15 mn delay. His experience suggests that the usual overall travel time (including the wait at the bus stop) is 1 hour. But if Tom is lucky (no wait, no traffic jam) it can take only 45 mn. and, if unlucky, 1 hour and 15mn. This information builds up a triangular fuzzy number T that models the uncertainty about the travel time; the membership function of T does not reflect preference. Tom has preference on his arrival time: he would like to arrive at 8 or before, but certainly not after 8:15 a.m..This makes up the fuzzy arrival time D. How can Tom choose his actual starting time?

Assuming that R, T, and D are intervals, what Tom wants to find is a starting time s that does not force him to get up too early (hence $s \in R$) such that whatever the travel time (hence for any $t \in T$) he arrives on time at the meeting ($s+t \in D$). Mathematically this means:

$\exists$ s such that: $s \in R$ and $\forall t$, if $t \in T$ then $s+t \in D$

When R, T, D are fuzzy sets it comes down to finding s that maximizes

$$\min(\mu_R(s), \inf_t \max(1- \mu_T(t), \mu_D(s+t)))$$

The maximization and minimization are the fuzzy counterpart of the universal ($\forall$) and existential ($\exists$) quantifiers; and max(1-a, b) is a multiple-valued implication. Clearly, the term $\inf_t \max(1- \mu_T(t), \mu_D(s+t))$ is equal to the degree of necessity of the fuzzy event $s+t \in D$ given the uncertain information $t \in T$, that is, $N(t \in (-\infty, D \ominus s])$ as in equation (23). Note that applying possibility theory, as in the previous subsection, gives a systematic method for handling uncertainty along with preference.

The problem would be quite different if Tom had control on his travel time (say he takes his car, and there is no traffic jam before 8 a.m.). Then the travel time t becomes controllable and T represents Tom's preference about his drive (not to be above speed limit, driving not too slow either, etc.). The problem becomes one of finding both s and t which maximize preferences about starting time, driving time, and arrival hour, i.e. $\max_{s,t} \min(\mu_R(s), \mu_T(t), \mu_D(s+t)))$, quite a different problem from the case when t is uncertain. This problem with preference only is trivially solved by letting s = 7 and t = 1 with degree of preference 1. The other problem looks more tricky, and is solved in the next section.

4.4. A unique formulation handles flexible as well as imprecise durations

When durations are decision parameters subject to flexible constraints, the different types of constraints over the possible values of the starting times (see equation (18)) can be expressed by:

precedence constraints $P_{i\to k}$:	$s_k - s_i \in [T(i), +\infty)$
capacity constraints $C_{i\leftrightarrow k}$:	$s_k - s_i \in [T(i), +\infty)$ or $s_i - s_k \in [T(k), +\infty)$
release and due dates:	$s_i \in [R(i), D(i) \ominus T(i)]$

The scheduling problem defines a non-conjunctive constraint graph whose nodes represent the operations. Precedence and temporal constraints define the conjunctive part of the graph: a fuzzy temporal window [R(i),T(i)] is associated to each node. Each conjunctive edge $P_{i\to k}$ represents a precedence constraint "O_i must precede O_k", by means of a fuzzy inequality of the type $s_k - s_i \geq T(i)$. Capacity constraints $C_{i\leftrightarrow k}$ define non-conjunctive edges ($s_k - s_i \geq T(i)$ OR $s_i - s_k \geq T(k)$) which represent conflicts of the type "O_i before O_k OR O_k before O_i".

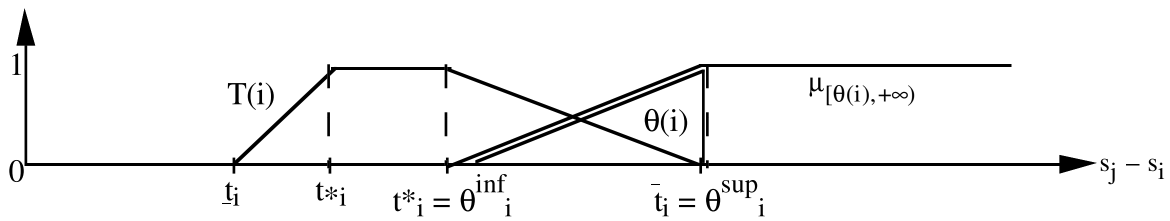

Figure 4: from]T(i), +∞) to [θ(i), +∞)

This kind of graph can also represent constraints involving ill-known durations. Indeed, in this case, a precedence constraint $P_{i\to k}$ requires that $s_k - s_i$ belongs to $]T(i),+\infty)$ the set of values which are *necessarily* greater than T(i), or, in other terms, that $s_k - s_i$ belongs to $[\theta(i),+\infty)$, the set of values that are possibly greater than θ(i), θ(i) being defined by $\theta^{inf}_i = \sup \text{core}(T(i)) = t^*_i$, $\theta^{sup}_i = \bar{t}_i$, $\mu_{\theta(i)}(t_i) = 1 - \mu_{T(i)}(t_i)$ for $t^*_i \leq t_i \leq \bar{t}_i$. Then $\mu_{[\theta(i),+\infty)} = \mu_{]T(i),+\infty)}$ (see Figure 4).

Now it is easy to see that, if Π' is the possibility measure computed with (3) with A = θ(i) and $P = (-\infty, s_k - s_i]$:

$$N(t_i \in (-\infty, s_k - s_i]) = \mu_{[\theta(i),+\infty)}(s_k - s_i) = \Pi'(t_i \in (-\infty, s_k - s_i]).$$

A little less obvious is the following result (which holds provided some continuity assumption on the membership functions) (see Appendix I)

$$N(t_i \in (-\infty, D(i) \ominus s_i] = \sup_{t_i} \min(\mu_{[\theta(i),+\infty)}(t_i), \mu_{(-\infty,D(i)\ominus s_i]}(t_i)) \quad (25)$$
$$= \Pi'(t_i \in (-\infty, D(i) \ominus s_i]).$$

Hence the new form of the following constraints where durations are not controlable:

precedence constraints $P_{i\to k}$: $s_k - s_i \in [\theta(i), +\infty)$
capacity constraints $C_{i\leftrightarrow k}$: $s_k - s_i \in [\theta(i), +\infty)$ or $s_i - s_k \in [\theta(k), +\infty)$
release and due dates: $s_i \in [R(i), D(i) \ominus \theta(i)]$

Hence, a scheduling problem with uncertain durations can be formally expressed by the same kind of constraints as a problem involving flexible durations, and thus be described by means of the same kind of non conjunctive graph. But the interpretation is quite different: in case of flexible durations, the fuzzy duration labels over the graph come from the specification of preferences and represent the possible values that can be assigned to the t_i's. In case of imprecisely known durations, these labels come from the uncertainty about the real value of some durations: each of them represents the set of values that are necessarily greater than the estimated duration of an operation. Note that in case of flexible durations only the increasing part of T(i) is used (since it is interesting only to shorten this duration in order to make the scheduling problem more feasible), while in the case of uncertain durations, only the decreasing part of T(i) is used, under the form θ(i) (since the decision-maker has to be pessimistic about the value of t_i in order to protect the schedule against hazards). In the following, it is supposed that fuzzy durations T(i) represent flexible durations, considering that imprecise durations can be handled, replacing T(i) by θ(i).

Example: Let us solve the mixed uncertainty/preference problem of Section 4.3. The fuzzy travel time T is changed into θ, with a lower modal value $\theta^{\inf} = 1.25$ and lower support limit $\theta^{\inf} = 1$. From the above result,

$$\sup_s \min(\mu_R(s), \inf_t \max(1- \mu_T(t), \mu_D(s+t))) = \sup_{s,\, t} \min(\mu_R(s), \mu_\theta(t), \mu_D(s+t)))$$
$$= \sup_s \min(\mu_R(s), \mu_{D \ominus \theta}(s)))$$

where using fuzzy arithmetic, D ⊖ θ has a linear increasing membership function, with upper modal value (in hours) 8 - 1.25 = 6.75, and upper support limit 8.25 - 1 = 7.25. Finding the optimum preference level is the matter of intersecting two straight lines (as in figure 5). The reader can check that this level is .75, and that the optimal time for Tom to start safely enough is 6.875 = $\mu_R^{-1}(.75)$ (= 6: 52' 30" a.m.). This result presupposes that the travel time does not exceed 1.1875

= $\mu_\theta^{-1}(.75)$ (1h 11' 20"); note that this is very likely. Moreover taking these conditions for granted, Tom will arrive at his meeting not later than $\mu_D^{-1}(.75) = 8.0625$ (= 8: 03' 45" a.m.). Note that this solution differs from the one when all fuzzy sets mean preference, and sounds quite reasonable in practice.

5. Solving flexible job-shop scheduling problems

The solving approach consists in searching for a sequencing of operations on each machine (like in most approaches), from which earliest and latest starting times can be computed. In other terms, as pointed out in Section 2, disjunctive constraints "O_i before O_k OR O_k before O_i" must be transformed into simple precedence constraints by choosing one of the alternatives. It is based on three basic procedures:

- a general search procedure that proceeds by managing new precedence constraints,
- a consistency enforcing procedure which propagates the effect of these decisions through the updated precedence (conjunctive) graph,
- a constraint analysis procedure (or look-ahead analysis) that determines which precedence decision to make next, i.e., it generates precedence constraints by solving disjunctions.

5.1. Consistency Enforcing

Considering only the conjunctive part of the graph, each new precedence constraint $P_{i\to k}$ implies that the fuzzy temporal window associated to O_k (resp. O_i) must now be such that $r_k \geq r_i + t_i$ (resp. $d_i \leq d_k - t_k$). Hence, the temporal windows can be updated as follows:

$$R'(k) := \widetilde{\max}(R(k), R(i) \oplus T(i))$$
$$D'(i) := \widetilde{\min}(D(i), D(k) \ominus T(k))$$

where $\widetilde{\max}$ and $\widetilde{\min}$ are versions of the maximum and minimum operations extended to fuzzy arguments. This is a variant of the initial computation of fuzzy temporal windows as in (10)-(11). The acyclicity of the graph can be exploited to produce an efficient ordering for updating the temporal windows. The R(k) are updated along precedence constraints, and the D(i) are updated backwards.

It turns out that this method guarantees that the best among the earliest (resp. latest) starting times according to the precedence constraints can then be obtained when assigning to each s_i the lowest (resp. greatest) date among its best possible values, i.e., values s^*_i with highest membership degree in the set $[R(i), D(i) \ominus T(i)]$ (see Figure 5).

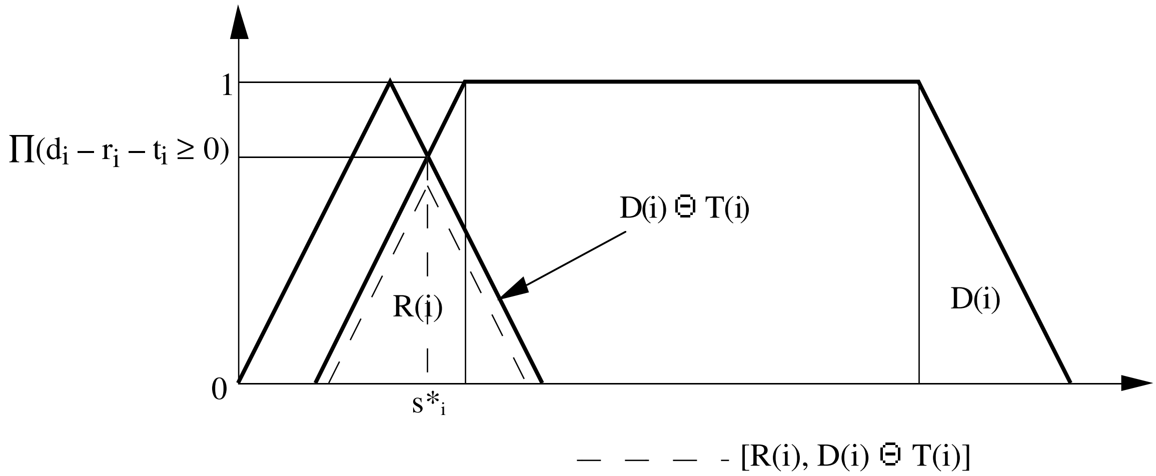

Figure 5: consistency degree of O_i's temporal window

Hence, the consistency of the conjunctive part of the problem, which is the satisfaction degree of the best scheduling according to the precedence and limit date constraints, is given by:

$$\begin{aligned} \text{Cons(conjunctive part)} &= \min_{i=1,n} \sup_{s_i} \min(\mu_{[R(i),+\infty)}(s_i), \min_{i=1,n} \mu_{(-\infty,D(i)\Theta T(i)]}(s_i)) \\ &= \min_{i=1,n} \Pi(d_i - r_i - t_i \geq 0). \end{aligned} \quad (26)$$

Cons(conjunctive part) considers only the precedence constraints, without taking into account the capacity constraints. It only yields an upper bound of the consistency of the global scheduling problem. However, Cons(conjunctive part) = 0 means that a contradiction is detected: constraints are totally inconsistent.

5.2. Search Procedure

The sequencing which is searched for is one of those having the best satisfaction degree.The search procedure can be in fact a classical branch &bound algorithm using a depth-first strategy. The nodes of the tree represent partial sequencing and its leaves complete sequences on machines: extending a node means choosing a disjunction (O_i precedes O_k OR O_k precedes O_i) and selecting one of its unexplored alternatives. This choice is done by the look-ahead procedure. The graph is then modified according to this decision (the corresponding linear inequality is substituted to the disjunction) and the consistency of the conjunctive part is enforced using the previous propagation algorithm, propagating D(k) backwards and R(i) forwards through the new edge in the precedence graph. Hence an estimate of Cons(conjunctive part) is obtained and associated to the node: this degree is an upper bound of the satisfaction degree of the best complete sequencing that can be reached from the node. A bound α represents the satisfaction degree of the best current complete sequencing (initialized to value 0): only nodes with satisfaction degree greater than α should be explored. If the current node is such that Cons(conjunctive part) $\leq \alpha$, the algorithm backtracks to a node whose degree is greater than α. α is updated each time a complete sequencing better than the previous one is reached.

This kind of algorithm clearly has a worst case behaviour not worse than that of classical backtracking used to solve crisp scheduling problems. If the search time is limited, the use of a depth first strategy allows to quickly obtain a sub-optimal solution which will be enhanced according to the remaining time. Moreover, the flexibility of the constraints is used to guide the search and allows the pruning of useless branches. Finally, it is possible to develop a large class of search algorithms (e.g., beam search as in Fox and Strohm (1982)) based on the same principles and integrating different variants (see Schiex, 1992).

5.3. Look ahead procedure

The efficiency of the search relies of course on the heuristic evaluation function that determines which disjunction to be instanciated next. It is actually based on the extension of constraint analysis tests (Erschler et al., 1976; Erschler and Esquirol, 1986) that have been described in Section 2. For each disjunction, also called a conflict (O_i precedes O_k OR O_k precedes O_i), an upper bound of the possibility of each alternative may be computed (Dubois, 1989):

$$\begin{aligned}\Pi_{sup}(O_i \text{ precedes } O_k) &= \Pi((d_k - r_i) - (t_i + t_k) \geq 0) \\ &= \sup_{x \geq 0} \mu_{D(k) \ominus R(i) \ominus T(i) \ominus T(k)}(x) \\ &= \mu_{(-\infty, D(k) \ominus R(i)]}(t_i + t_k) \text{ in case of crisp durations}\end{aligned} \tag{27}$$

This is the fuzzy version of Test-1(i,k) described in Section 2. $\Pi_{sup}(O_i$ precedes $O_k) = 0$ means that the decision "O_i precedes O_k" is inconsistent in the current search state according to O_i and O_k's temporal windows. Hence, decision "O_k precedes O_i" must be chosen (otherwise, the satisfaction degree of the sequencing will be 0). Note that the the calculation $\Pi_{sup}(O_i$ precedes $O_k)$ according to (27) only gives an upper bound of $\Pi(O_i$ precedes $O_k)$. For instance, it would be inefficient in the three-operation example of Section 2. An approximation of $\Pi_{sup}(O_i$ precedes $O_k)$ better than the one computed by (27) can be obtained taking three operations into account, say O_i, O_k and O_x, and letting O_x vary, i.e. using a fuzzy extension of Test-2(i,k) described in Section 2:

$$\begin{aligned}\Pi_{sup}(O_i \text{ precedes } O_k) = \min(\ & \Pi((d_k - r_i) - (t_i + t_k) \geq 0), \\ & \min_{O_x} (\max\ \Pi((d_k - r_i) - (t_i + t_k + t_x) \geq 0), \\ & \qquad \Pi((d_k - r_x) - (t_i + t_k + t_x) \geq 0), \\ & \qquad \Pi((d_x - r_i) - (t_i + t_k + t_x) \geq 0)))\end{aligned} \tag{28}$$

where the three additional terms pertain to the respective sequences: $O_i/O_x/O_k$, $O_x/O_i/O_k$ and $O_i/O_k/O_x$. Coming back to the three-operation example of Section 2, $\Pi_{sup}(O_i$ precedes $O_k) = 0$ is obtained, which enforces the other precedence constraint.

Note that max(Π_{sup}(O_i precedes O_k), Π_{sup}(O_k precedes O_i)) is an upper bound of the satisfaction degree of the search state. Hence, backtracking can be caused by the look-ahead analysis as soon as there is a conflict such that max (Π_{sup} (O_i precedes O_k), Π_{sup}(O_k precedes O_i)) $\leq \alpha$, α being the satisfaction degree of the current best instantiation. Moreover, min(Π_{sup}(O_i precedes O_k), Π_{sup}(O_k precedes O_i)) estimates the degree to which the satisfaction will fall down if the best of the two alternatives is not chosen. The interest of choosing the best alternative between $P_{i \to k}$ and $P_{k \to i}$, that is, the criticity of the conflict (O_i , O_k), is defined by:

$$C(O_i,O_k) = 1 - \min(\Pi_{sup}(O_i \text{ precedes } O_k), \Pi_{sup}(O_k \text{ precedes } O_i)) \tag{29}$$

The criticity of a resource is the maximal criticity of the conflicts between operations to which the resource has been assigned. It evaluates the degree of necessity that either O_i should precede O_k or O_k should precede O_i.

Hence, the look ahead procedure first computes the criticity of each conflict. Not all of the remaining conflicts are analyzed at each search state, but only those involving at least one operation whose fuzzy temporal window has been modified by the consistency enforcing procedure while creating the search state (the criticity of the others remain unchanged). In order to keep the satisfaction degree as high as possible, a conflict whose criticity is maximal is then chosen to be instanciated by the most possible alternative (alternatives (O_i precedes O_k) such as Π_{sup}(O_i precedes O_k) $\leq \alpha$ will lead to a solution worse than the best current one: they do not have to be envisaged). If several conflicts correspond to the maximal criticity, one can focus on the machine having the larger set of critical conflicts. Since the set of the critical conflicts pertaining to a machine is a fuzzy set, this cardinality is actually a fuzzy cardinality (sum of membership grades).

Although the present framework is not additive like the one proposed in (Sadeh, 1991), its heuristics are similar: the lower the quality of the best scheduling deriving from the choice of a precedence constraint, the higher the priority of the conflict (and of the opposite decision). It is quite different from the heuristic chosen in the fuzzy approach described in (Kerr and Walker, 1989), in which the most possible decisions are chosen regardless of their degrees of criticity. In that work, the priority of a conflict is the maximum possibility of the alternatives: a conflict involving two completely possible alternatives is paradoxically considered as interesting to the same extent as a conflict involving one impossible alternative and a completely possible one. The approach described here considers the latter conflict as more interesting and enforces the only feasible decision. An illustrative example solved by the implemented method is given in Appendix II.

The look-ahead procedure can be linked with a knowledge-based decision support module (see for instance Bensana et al. (1988)): on the one hand, the knowledge-based module can be used to break ties among otherwise equivalent candidates. On the other hand, there are situations where the most critical conflicts are not critical enough to efficiently motivate a decision: even the worst alternative will not decrease the satisfaction degree of the next node (because the highest criticity is lower than 1 – Cons(conjunctive part of the current node)). It is then better to use other criteria to solve the conflicts, especially knowledge-based criteria (which can also exploit the possibility degrees computed by the constraint analysis procedure). In the OPAL system (Bensana et al., 1988), the constraint propagation is not fuzzy but is exactly the one of Erschler et al. (1976); fuzzy sets are only used in the representation of various priority rules which are triggered in order to compute preference profiles over precedence decisions. These preference profiles are combined using a voting procedure that extends the majority rule. More elaborated aggregation schemes are proposed by Dubois and Koning (1994).

6. Conclusion

Possibility theory offers a rich and powerful setting for the representation of scheduling constraints pervaded with flexibility (e.g., flexible release dates, due dates and durations) or uncertainty (e.g., imprecise durations). Classical constraint propagation (e.g., scheduling in a conjunctive activity graph) and constraint analysis schemes can be easily extended to this new framework. In any case, a fuzzy approach reveals suitable for scheduling problems involving relaxable constraints or imprecise limit dates: explicitly taking the flexibility of the problem into account does not significantly change the computational cost of the search procedure; the complexity of consistency enforcing and analysis procedures may be multiplied by two in the worst case. Moreover, empirical relaxations techniques are avoided since they often happen to be more expensive, difficult to formulate, and suboptimal. This framework can be easily extended to capture priority between constraints (e.g., "it is preferable to schedule O_i before O_k but it is not compulsory") see (Dubois et al., 1994). It should also be noticed that the fuzzy approach can handle partially inconsistent problems. A solution (the instanciation with the maximal satisfaction degree) will be provided as long as the problem is not totally inconsistent.

Preliminary experiments had been made (Mathé, 1987) to compare fuzzy constraint analysis involving flexible release dates and due dates to crisp constraint analysis in the framework of the Opal system (Bensana et al., 1988). Except for strongly constrained problems where a crisp constraint analysis can make most of the decisions, these preliminary experiments suggested that the fuzzy analysis might be more productive than the crisp analysis. The integration of fuzzy durations has now been implemented and its computational comparison to crisp durations for real-sized problems is currently performed. Moreover, the more elaborate look-ahead scheme (28) is currently under experiments to determine whether it really enhances the predictiveness of the search procedure and it significantly reduces the search space (improving the estimation of the possibility of each alternative, the analysis will lead to more pruning during the search but each step will be more time consuming). Results obtained so far are encouraging (Fargier, 1994). They indicate that computational times to solve flexible jobshop scheduling problems are not significantly greater than the times requested to solve hard, non-fuzzy ones, regarding to finding a first feasible solution. Sometimes fuzzy jobshops get solved even faster. Moreover the first solution found using the fuzzy formulation is almost systematically better in terms of quality: starting times are chosen well inside the feasibility domains while the first solutions to crisp formulations tend to stick to the boundaries of the feasibility domain. The solutions computed by the procedure depend on the choice of the membership functions, of course. However the sensitivity should not be too dramatic, because due to the use of the minimum operation a small change in the membership functions will be either directly reflected in the global satisfaction level or ignored. In this paper, an approach has been presented for computing predictive schedules taking into account flexibilities and uncertainty. Clearly, one may also think of taking advantage of fuzzy constraints in a real-time monitoring perspective.

Appendix I

1) Proof of the identity (17)**:**

$\mathrm{Sat}(s_1, \ldots, s_n) = \min_{i=1,n} \min(\min_{k\in \mathcal{S}_i} \mu_{[T_i,+\infty)}(s_k - s_i), \mu_{R(i)}(s_i), \mu_{D(i)\ominus T(i)}(s_i))$.

One must prove that, in this particular situation, the sup distributes over the min, that is,

$$\sup_{t_i} \min(\mu_{(-\infty,\min_{k\in \mathcal{S}_i} s_k]}(s_i + t_i), \mu_{T(i)}(t_i), \mu_{(-\infty,D(i)]}(t_i + s_i)) =$$
$$\min(\sup_{t_i} \min(\mu_{(-\infty,\min_{k\in \mathcal{S}_i} s_k]}(s_i + t_i), \mu_{T(i)}(t_i)), \sup_{t_i} \min(\mu_{T(i)}(t_i), \mu_{(-\infty,D(i)]}(t_i + s_i))). \quad \text{(A1)}$$

Indeed if this equality is established, then

$$\sup_{t_i} \min(\mu_{(-\infty,\min_{k\in \mathcal{S}_i} s_k]}(s_i + t_i), \mu_{T(i)}(t_i)) = \sup_{t_i \le (\min_{k\in \mathcal{S}_i} s_k) - s_i} \mu_{T(i)}(t_i)$$
$$= \mu_{[T(i),+\infty)}(\min_{k\in \mathcal{S}_i} s_k - s_i) = \min_{k\in \mathcal{S}_i} \mu_{[T(i),+\infty)}(s_k - s_i).$$

The other term has been computed in equation (16).

To establish the distributivity of the sup in (A1) the following lemma is needed:

Lemma: If A, B, C are three continuous fuzzy intervals then $((-\infty,A] \cap (-\infty,B]) \ominus C = (-\infty, A \ominus C] \cap (-\infty, B \ominus C]$.

Proof: Let $f = \mu_{(-\infty,A]}$ and $g = \mu_{(-\infty,B]}$. f and g are decreasing functions on $\mathbb{R}$. Namely $f(x) = 1$, $\forall\, x \le a^*$, $f(x) = 0$, $\forall\, x \in [\bar{a},+\infty)$, f is decreasing on $[a^*,\bar{a}]$. The same holds for g with respect to b^* and $\bar{b}$. $\mu_C(x) = 0$ for $x \notin [\underline{c},\bar{c}]$, $\mu_C(x) = 1$ for $x \in [c_*,c^*] \subseteq [\underline{c},\bar{c}]$, μ_C is increasing on $[\underline{c},c_*]$, and decreasing on $[c^*,\bar{c}]$. Now, it is always true that:

$$\mu_{((-\infty,A]\cap(-\infty,B])\ominus C}(z)= \sup_x \min(f(x + z), g(x + z), \mu_C(x))$$
$$\le \min(\sup_x \min(f(x + z), \mu_C(x)), \sup_x \min(g(x + z), \mu_C(x))). \quad \text{(A2)}$$

Let x_f and x_g be such that $\sup_x \min(f(x + z), \mu_C(x)) =\min(f(x_f + z), \mu_C(x_f))$ and $\sup_x \min(g(x + z), \mu_C(x)) = \min(g(x_g + z), \mu_C(x_g))$. Assume $f(x_f + z) = \mu_C(x_f) = \mu_C(x_g) = g(x_g + z) = 1$, and without loss of generality $x_f \le x_g$; then $g(x_f + z) = 1$, and x_f maximizes $\min(f(x + z), g(x + z), \mu_C(x))$ as well. Changing 1 into 0, the equality obviously holds in (A2). Assume now that $\min(f(x_f + z), \mu_C(x_f)) \in (0,1)$. Then it is obvious that $x_f \in (\underline{c}, c_*)$, and that $f(x_f + z) = \mu_C(x_f) < 1$. Assume $f(x_f + z) < g(x_g + z)$. Hence $\mu_C(x_f) < \mu_C(x_g)$ and $x_f < x_g$. Hence $g(x_f + z) > g(x_g + z) > f(x_f + z)$. Hence

$$\min(\mu_C(x_f), f(x_f + z), g(x_f + z)) = \min(\mu_C(x_f), f(x_f + z))$$

and x_f is a maximum for $\min(\mu_C(x), f(x + z), g(x + z))$. Assume now $f(x_f + z) = g(x_g + z)$. Without loss generality assume $x_f \le x_g$ then $g(x_f + z) \ge g(x_g + z) = f(x_f + z)$. Again x_f is a maximum for $\min(\mu_C(x), f(x + z), g(x + z))$.

Now it is enough to prove that $\mu_{(-\infty,A]\ominus C}(z) = \mu_{(-\infty,A\ominus C]}(z)$.

To see it notice that

$$\begin{aligned}
\mu_{(-\infty,A]\ominus C}(z) &= \sup_x \min(\mu_C(x), \sup_{y\ge x+z} \mu_A(y)) \\
&= \sup_{x,y:y\ge x+z} \min(\mu_C(x), \mu_A(y)) \\
&= \sup_{t\ge z} \sup_x \min(\mu_C(x), \mu_A(t + x)) \text{ with } t = y - x \\
&= \sup_{t\ge z} \mu_{A\ominus C}(t) = \mu_{(-\infty,A\ominus C]}(t).
\end{aligned}$$

2) Proof of the identity (25)

$$N(t_i \in (-\infty, D(i) \ominus s_i]) = \sup_{t_i} \min(\mu_{[\theta_i,+\infty)}(t_i), \mu_{(-\infty,D(i)\ominus s_i]}(t_i)).$$

Let $f = \mu_{(-\infty,D(i)\ominus s_i]}$, $g^- = \mu_{(-\infty,T(i)[}$, $g^+ = \mu_{]T(i),+\infty)} = \mu_{[\theta_i,+\infty)}$. It holds $\mu_{T(i)} = \min(1-g^+, 1-g^-)$. f is continuous, decreasing in the wide sense, as well as g^-, while g^+ is continuous, increasing in the wide sense. Moreover, $g^-(x) = 0$, $\forall\, x > t_{*i}$, $g^+(x) = 0$, $\forall\, x < t^*_i$. (refer to fig.4).

Let $t \in (t_{*i}, t^*_i)$. Then

$$\begin{aligned}
N(t_i \in (-\infty, D(i) \ominus s_i)) &= \inf_x \max(f(x), g^+(x), g^-(x)) \\
&= \min(\inf_{x\le t} \max(f(x), g^-(x)), \inf_{x\ge t} \max(f(x), g^+(x))) \\
&= \min(\max(f(x_*), g^-(x_*)), \max(f(x^*), g^+(x^*)))
\end{aligned}$$

for some $x^* \ge t$, and $x_* \le t$. If $\max(f(x^*), g^+(x^*)) = 1$ it means that $\bar{t_i} \le d^{inf}{}_i - s_i$, hence $f(x) = 1$ for all $x \le \bar{t_i}$. Hence $N(t_i \in (-\infty, D(i) \ominus s_i]) = \inf_x \max(f(x), g^+(x)) = 1$. If $\max(f(x^*), g^+(x^*)) = 0$, then $N(t_i \in (-\infty, D(i) \ominus s_i]) = \inf_x \max(f(x), g^+(x)) = 0$. If $\max(f(x^*), g^+(x^*)) \in (0,1)$ then because f is decreasing while g^+ is increasing in the neighborhood of x^*, clearly $f(x^*) = g^+(x^*)$. But $\max(f(x_*), g^-(x_*)) \ge f(x^*)$ since f is decreasing. Hence in all cases it holds that

$$N(t_i \in (-\infty, D(i) \ominus s_i]) = \inf_x \max(f(x), g^+(x)).$$

Now due to the continuity of f and g+, it is easy to check that

$$\inf_x \max(f(x), g^+(x)) = \sup_x \min(f(x), g^+(x)).$$

When T(i) is a regular closed interval g^- and g^+ are not continuous but it can be verified that the result holds.

References

Bellman R.E., Esogbue A.O., Nabeshima I. (1982) *Mathematical Aspects of Scheduling and Applications*. Pergamon Press, Oxford, U.K

Bellman R.E. Zadeh, L.A.(1970) Decision-making in a fuzzy environment. Management Science, 17(4), B141-B164

Bensana E., Bel G., Dubois D. (1988), "OPAL, a multi-knowledge-based system for industrial job-shop scheduling", *Int. J. of Production Research* , 26(5), 795-816.

Blackstone J.H., Phillips D., Hogg G.L. (1982) A state-of-the-art survey of dispatching rules for manufacturing job-shop operations, *Int. J. of Production Research* , 20(1)

Chanas S., Kamburowski J. (1981), "The use of fuzzy variables in PERT", *Fuzzy Sets and Systems*, 5, 11-19.

Davis E.(1987) Constraint propagation with interval labels, *Artificial Intelligence* , 32, 281-331

Dechter R., Meiri I., Pearl J. (1991), "Temporal Constraint Networks", *Artificial Intelligence*, 49, 61-95.

Descottes Y., Latombe J.C. (1985), "Making compromises among antagonist constraints", *Artificial Intelligence* , 27, 159-174.

Dubois D. (1987), "An Application of Fuzzy Arithmetics to the Optimization of Insustrial Machining Processes", *Mathematical Modelling* 9, 461-475.

Dubois D. (1989), "Fuzzy knowledge in an artificial intelligence system for job-shop scheduling", in: G. Evans, W. Karwowski and M. Wilhelm (eds.), *Applications of the Fuzzy Set Methodologies in Industrial Engineering,* Elsevier, 73-89.

Dubois D., Fargier H., Prade H. (1993) "Flexible constraint satisfaction problems with application to scheduling problems", Report IRIT/93-30-R, I.R.I.T., Univ. P. Sabatier, Toulouse.

Dubois D., Fargier H., Prade H. (1994), "Propagation and satisfaction of flexible constraints", in: R.R. Yager and L.A. Zadeh (eds.), *Fuzzy Sets, Neural Networks and Soft Computing*, Kluwer Academic, Dordrecht.

Dubois D., Koning J.L.(1994), "A decision engine based on rational aggregation of heuristic knowledge", Decision Support Systems, 11, 337-361

Dubois D., Prade H. (1980), *Fuzzy Sets and Systems: Theory and Applications.* Academic Press, New York.

Dubois D., Prade H. (with the collaboration of Farreny H., Martin-Clouaire R., Testemale C.) (1988), *Possibility Theory — An Approach to Computerized Processing of Uncertainty,* Plenum Press, New York.

Dubois D., Prade H. (1989), "Processing fuzzy temporal knowledge", *IEEE Trans. on System, Man and Cybernetics*19(4),729-744.

Erschler J. F., Roubellat J.P., Vernhes (1976), "Finding some essential characteristics of the feasible solutions for a scheduling problem", *Operations Research* 24, 774-783.

Erschler J., Esquirol P. (1986), "Decison-aid in job shop scheduling: A knowledge based apporach", *Proc. of the 1986 IEEE Conf. on Robotics and Automation*, San Franscisco, 1751-1756.

Erschler J., Lopez P., Thuriot C. (1989), "Temporal reasonning under resources constraints: Application to task scheduling", *Proc. of the 2nd Inter. Symp. on Systems Research, Informatics and Cybernetics*, Baden-Baden.

Erschler J., Lopez P., Thuriot C. (1991), "Raisonnement temporel sous contraintes de ressource et problèmes d'ordonnancement", *Revue d'Intelligence Artificielle* 5(3), 7-32.

Fargier H. (1994) *Problèmes de satisfaction de contraintes flexibles: application à l'ordonnancement de production* (In French). Ph.D. Thesis, Université Paul Sabatier, Toulouse, 314p.

Fox M.S. (1987), *Constraint-Directed Search: A Case Study of Job-Shop Scheduling*. Pitman, London.

Fox M.S., Smith S.F. (1984), "ISIS: A knowledge based system for factory scheduling", *Expert Systems* 1(1), 25-49.

Fox M.S., Strohm G.A. (1982), "Job-shop scheduling: An investigation in constraint-directed reasoning", *Proc. of the. Natioonal Conf. on Artificial Intelligence (AAAI'82)*, Pittsburgh.

Fox M.S., Zweben, M., Eds. (1993) *Knowledge-based Scheduling*, Morgan and Kaufmann, San Mateo, Ca.

Freuder E.C. (1992), "Partial constraint satisfaction", *Artificial Intelligence*, 58, 24-71.

Freuder E.C. Snow P. (1990) Improved relaxation and search methods for approximate constraint satisfaction with a maximin criterion. *Proc. 8th Conf. of the Canadian Soc. For Computational Studies of Intelligence*, 227-230

Gazdik I. (1983), "Fuzzy network planning", *IEEE Trans. on Reliability* 32, 304-313.

Grabot, B. Geneste L. (1994) Dispatching rules in scheduling: a fuzzy approach. *Int. J. of Production Research* , 32(4), 903-915

Ishibushi H., Yamamoto N., Misaki. S., Tanaka H. (1994) Local search algorithms for flowshop scheduling with fuzzy due-dates. *Int. J. Production Economics* , 33, 53-66

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

Kerr R.M., Walker R.N. (1989), "A job shop scheduling system based on fuzzy arithmetics", *Proc. 3rd Inter. Conf. on Expert Systems and the Leading Edge in Production and Operation Management*, Hilton Head, SC, 433-450.

Laguna M., Barnes J.W., Glover F.W. (1991) Tabu search methods for a single machine scheduling problem, *J. Intelligent Manufacturing* , 2, 63-74.

Lhomme O. (1993) Consistency techniques for numeric CSP's. *Proc 13th Joint Int. Conf. on Artificial Intelligence* (IJCAI93), 231-237

Lootsma F.A. (1989), "Stochastic and fuzzy PERT", *Europ. J. of Operational Research* 43, 174-183.

Lepape C. (1985), "SOJA: A daily workshop scheduling system", *Proc. of the BCS Specialist Group on Expert Systems Conf.*, Warwick, 225-242.

Mackworth A.K. (1977), "Consistency in networks of relations". *Artificial Intelligence* 8, 99-121.

Mathé N. (1987), "Prise en compte de l'imprecision des delais dans la construction d'ordonnancements prévisionnels", Master Degree Thesis, Univ. P. Sabatier, Toulouse.

Montanari H. (1974), "Networks of constraints: Fundamental properties and application to picture processing", *Information Science* 7, 95-142.

Montazeri M. (1990), Analysis of scheduling rules for an FMS, *Int. J. of Production Research* , 28(4), 785-802

Nasution S.H. (1993), "Fuzzy critical path method", *IEEE Trans. on Systems, Man and Cybernetics* 24(1), 48-57.

Peng S.O., Smith S.F. (1986),"Toward an opportunistic scheduling system", *Proc. of the 19th Hawaï Inter. Conf. on Systems Science.*

Prade H. (1979), "Using fuzzy set theory in a scheduling problem: A case study", *Fuzzy Sets and Systems* 2, 153-165.

Sadeh N. (1991), "Look-ahead techniques for micro-opportunistic job shop scheduling", Report CS91-102, Carnegie Mellon Univ., Pittsburg.

Schiex T. (1992), "Possibilistic constraint satisfaction problems or how to handle soft constraints", *Proc. of the 8th Conf. on Uncertainty in Artificial Intelligence*, Stanford, July 19-22, 268-275.

Sakawa, M. (1993) *Fuzzy Sets and Interactive Multiobjective Optimization.* Plenum Press New-York

Slowinski, R. Teghem, J. (1990), *Stochastic versus Fuzzy Approaches to Multiobjective Linear Programming under Uncertainty*. Kluwer Academic Pub. Dordrecht, The Netherlands.

Van Hentenryck (1990) *Constraint satisfaction and logic programming*, the MIT press Cambridge, Mass.

Van Laarhoven P.J.M. Aarts EHL Lenstra J.K.(1994) Job-shop scheduling by simulated annealing, *Operations Research*, 40, 113-125

Wood K.L., Otto K.N., Antonsson E.K. (1992), "Engineering design calculations with fuzzy parameters", *Fuzzy Sets and Systems* 52, 1-20.

Zadeh L.A. (1975), The concept of a linguistic variable and its applications to approximate reasoning, Part I, Information sciences, 8, 301-357

Zadeh L.A. (1978), "Fuzzy sets as a basis for a theory of possibility", *Fuzzy Sets and Systems* 1, 3-28.

Zimmermann, H.J., 1976. Description and optimization of fuzzy systems. Int. J. of General Systems, 2, 209-215

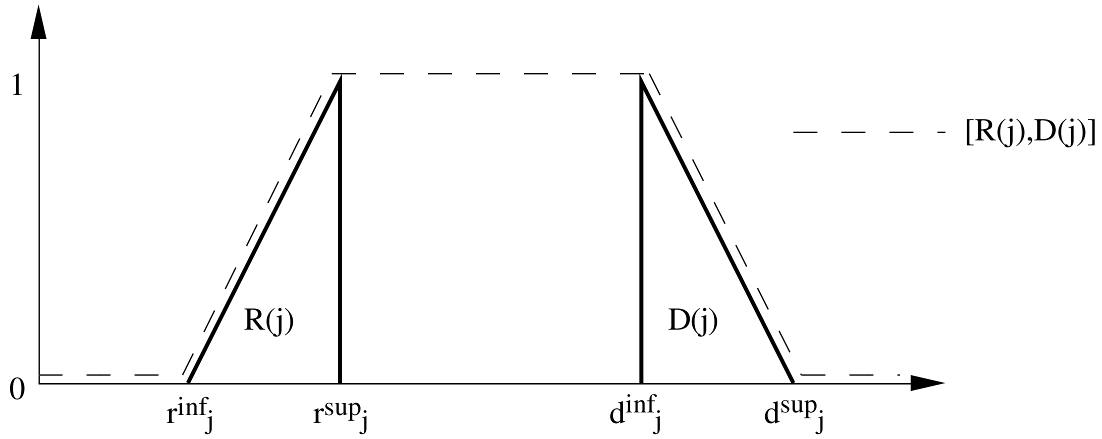

Figure 1: Fuzzy time horizon for job j

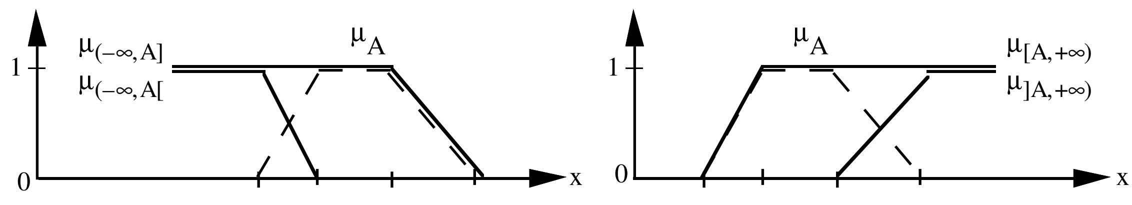

Figure 2: (a) points possibly/necessarily before A; (b) points possibly/necessarily after A

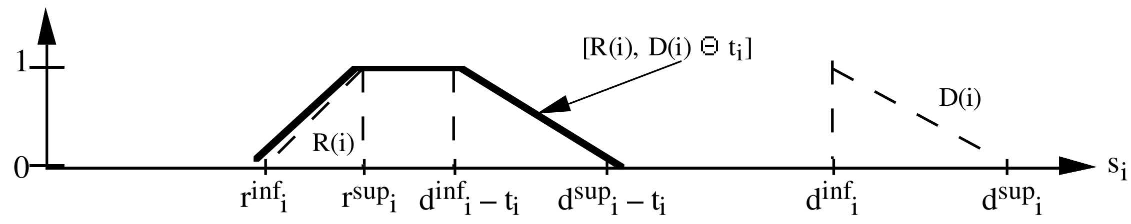

Figure 3: possible values for s_i given the temporal constraints on operation i.

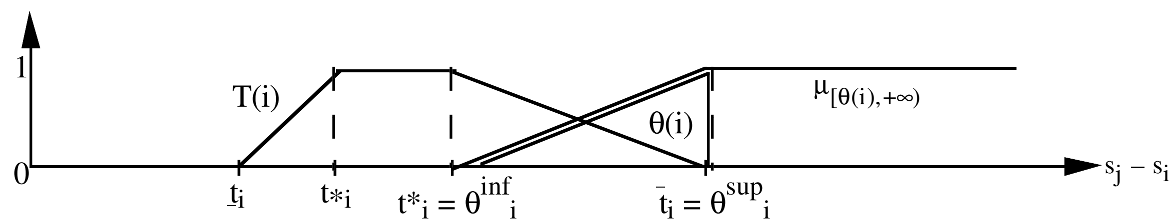

Figure 4: from]T(i), +∞) to [θ(i), +∞)

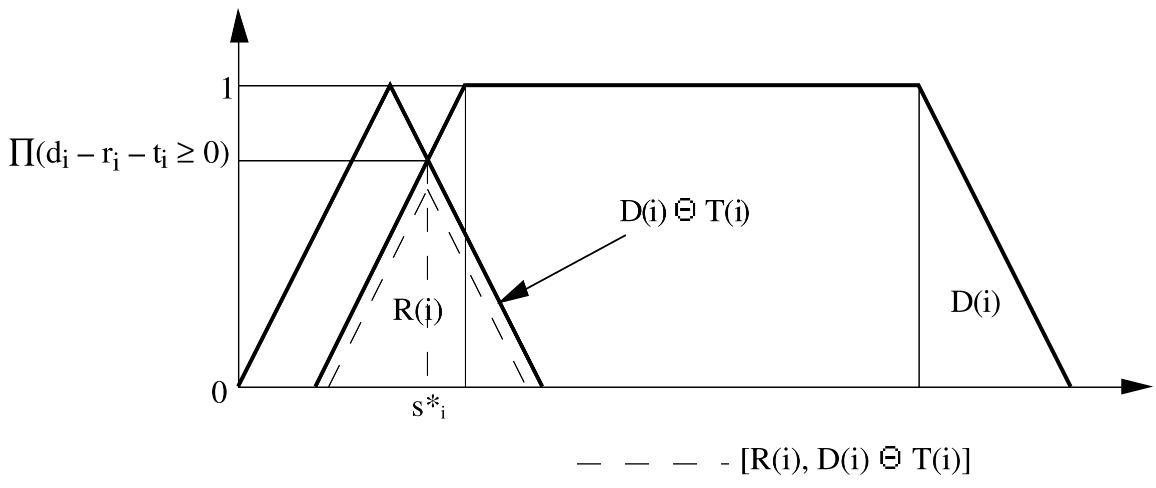

Figure 5: consistency degree of O_i's temporal window