



HAL
open science

The use of Fuzzy constraints in job-shop scheduling

Didier Dubois, H el ene Fargier, Henri Prade

► **To cite this version:**

Didier Dubois, H el ene Fargier, Henri Prade. The use of Fuzzy constraints in job-shop scheduling. IJCAI-93 Workshop on Knowledge-Based Production Planning, Scheduling and Control (W20 1993), Aug 1993, Chamb ery, France. pp.101-112. hal-04065921

HAL Id: hal-04065921

<https://hal.science/hal-04065921>

Submitted on 12 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destin ee au d ep ot et  a la diffusion de documents scientifiques de niveau recherche, publi es ou non,  emanant des  tablissements d'enseignement et de recherche fran ais ou  trangers, des laboratoires publics ou priv es.

KNOWLEDGE-BASED
PRODUCTION PLANNING,
SCHEDULING AND
CONTROL

→ W20

IFIP 93



INTERNATIONAL
JOINT CONFERENCE
ON ARTIFICIAL
INTELLIGENCE
CHAMBERY
SAVOIE FRANCE

The use of fuzzy constraints in job-shop scheduling

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

1. Introduction

Developing a good predictive schedule that satisfies temporal, technological and preference constraints is basically a search problem, the solution of which requires both powerful search heuristics and adequate means of representation (e. g. [1], [14], [17], [21], [22]). When scheduling over longer horizons, considering release date and due date constraints as compulsory may lead to reject an efficient schedule even when the violation is insignificant with regard to the precision of the realistic limits of predictability (e. g. in [16]). Significant computation effort may be actually saved by avoiding scheduling conflicts that are really insignificant.

On the one hand, temporal constraints prove often to be more or less relaxable or are subject to preferences (e. g. in [14] or [22]). As outlined in [5], [9], [16], fuzzy sets appear as a suitable framework for the representation of such flexible temporal constraints. A general approach to flexible Constraints Satisfaction Problems (CSPs) is described in [6]. On the other hand, 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 based on Zadeh's possibility theory [24][8] which can be understood as a fuzzy extension of the one proposed by [10]. This new framework allows us to handle flexible temporal constraints over release dates, due dates and durations, as well as uncertain durations.

The next section presents how fuzzy constrained scheduling problems can be defined in order to take into account flexible temporal constraints (release and due dates constraints) or fuzzy durations depending on their interpretation (flexible durations, which are under our control, or uncertain durations, which can only be fuzzily estimated). Section 3 then explains how fuzzy non conjunctive graphs of linear inequalities can deal with the representation of such constraints. In Sections 4, we present our solving scheme, which relies on three basic procedures: consistency enforcing, tree search and look-ahead analysis. Section 5 compares these procedures with solving paradigms proposed by the classical Constraint Satisfaction Problems (CSP) framework.

2. The fuzzily constrained scheduling problem

A 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 of operations according to a process plan that specifies a partial ordering among these operations (precedence constraints). 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 d_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. Denoting st_i the starting time of O_i , rd_i its release date and dd_i its due date, the different constraints translate into linear inequalities of the type:

precedence constraints $P_i \rightarrow K_i$ $st_k - st_i \geq dt_i$ (O_i before O_k)
capacity constraints $C_i \leftarrow K_i$ $st_k - st_i \geq dt_i$ or $st_i - st_k \geq dk_i$ (O_i and O_k cannot overlap),
release date constraints R_i : $st_i \geq rd_i$
due date constraints D_i : $st_i + dt_i \leq ddi_i$

Release and due dates of jobs are often subject to preferences. For instance job *j* must absolutely be completed at the latest completion date $dd_{sup}(j)$ (e. g. the date after which the customer will refuse delivery). Moreover it should preferably be completed before the due date $dd_{inf}(j)$ or as soon as possible after this due date. Similarly, it is better to begin the job after its preferred release date $rd_{sup}(j)$, because we are sure that the corresponding raw material will be available, while it is impossible to begin it before the earliest acceptable release date $rd_{inf}(j)$ ($rd_{inf}(j) \leq rd_{sup}(j) < dd_{inf}(j) \leq dd_{sup}(j)$). The knowledge about the release date (resp. due date) associated to *j* is no longer crisp but can be modeled by means of a fuzzy number $\tilde{rd}(j)$ (resp. $\tilde{dd}(j)$). Hence, the temporal window in which the job must take place is the fuzzy interval $[\tilde{rd}(j), \tilde{dd}(j)]$. Taking into account the precedence constraints, temporal constraints pertaining to the jobs define a fuzzy temporal window $[\tilde{rd}(i), \tilde{dd}(i)]$ for each operation O_i [5][9].

Let us first recall some results from possibility theory [8]. Consider a parameter *x* whose values are restricted by a fuzzy set A ($\pi_A(x) = \mu_A(x)$). The possibility of the event " $x \in P$ " denoted $\Pi(x \in P)$ is the intersection degree of *A* and *P*: $\Pi(x \in P) = \sup_u \min(\mu_A(u), \mu_P(u))$. It estimates to what extent " $x \in P$ " is possibly true, or if we prefer to what extent " $x \in P$ " is consistent with the information " $x \in A$ " modelled by $\pi_A(x) = \mu_A(x)$ (we make an extensive use of the sign " \in " since *P* or *A* may be fuzzy sets; it expresses a fuzzy belonging which is a matter of degree).

The necessity of " $x \in P$ " denoted $N(x \in P)$ measures to what extent *A* is included in the core of *P*, or, in other terms, to what extent " $x \in P$ " is certainly true, i.e. is entailed by " $x \in A$ ":

$$N(x \in P) = \inf_u \max(1 - \mu_A(u), \mu_P(u)).$$

In particular, if *x* is a real variable, Δ a fuzzy interval and *p* a crisp number and we have:

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

where $(-\infty, A]$, $(-\infty, A[$, $[A, +\infty)$, $]A, +\infty)$ denote respectively the set of points possibly before *A*, necessarily before *A*, possibly after and necessarily after *A* (see Fig. 1).

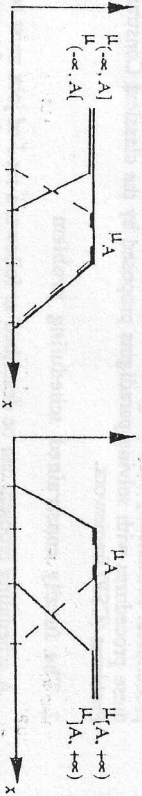


Figure 1: (a) points possibly/necessarily before *A*; (b) points possibly/necessarily after *A*

When the knowledge about the due date is fuzzy, the constraint $dd_i \geq st_i + di$ defined by this due date is not necessarily violated or satisfied but its satisfaction can be matter of degree. Indeed, the coefficient

$$\begin{aligned} \Pi(dd_i \geq st_i + di) &= \Pi(dd_i \in [st_i + di, +\infty)) \\ &= \mu_{(-\infty, \tilde{dd}(i)]}(st_i + di) \end{aligned} \quad (1)$$

can be understood as the satisfaction degree of the due date constraint for O_i starting at st_i : $\Pi(dd_i \geq st_i + di)$ indicates to what extent there exists a value for dd_i greater than $st_i + di$, given that dd_i is restricted by $\tilde{dd}(i)$: it is equal to 1 if $dd_{inf}(i) \geq st_i + di$, that is to say if the operation is

completed before the preferred due date. If O_i finishes after the latest acceptable completion date ($dd_{sup}(i) \leq st_i + di$) we have $\mu_{(-\infty, \tilde{dd}(i)]}(st_i + di) = 0$. Otherwise, the closer $st_i + di$ to the preferred due date, the higher $\mu_{(-\infty, \tilde{dd}(i)]}(st_i + di)$. 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 due date to the latest acceptable completion date. Note that $\Pi(dd_i \geq st_i + di)$ can also be viewed as the necessity that a precise st_i (which is fixed) belongs to the fuzzy set $[dd(i) \ominus di, +\infty)$.

Similarly, the release date of O_i being more or less relaxable from preferred release date to earliest possible one, the satisfaction degree of the release date constraint is:

$$\Pi(\tilde{rd}(i) \leq st_i) = \Pi(\tilde{rd}(i) \in (-\infty, st_i]) = \mu_{[\tilde{rd}(i), +\infty)}(st_i) \quad (2)$$

Hence, the set of more or less possible values for the starting time O_i according to the two flexible temporal constraints is the fuzzy interval $[\tilde{rd}(i), \tilde{dd}(i) \ominus di]$ (see Fig. 2).

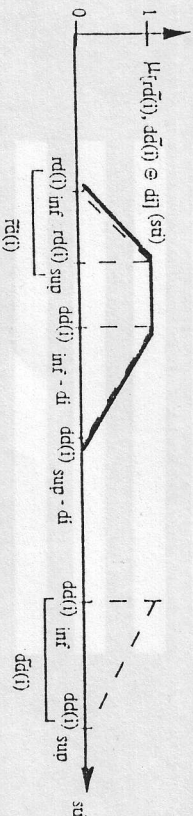


Figure 2: possible values for st_i given the temporal constraints over O_i

A solution to a crisp scheduling problem is typically an assignment (st_1, \dots, st_m) of starting times of all the operations that satisfies 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 $inscr$ as it satisfies the least satisfied temporal constraint. The global satisfaction level is defined as:

$$\begin{aligned} \text{Sat}(st_1, \dots, st_m) &= 0 \text{ if } (st_1, \dots, st_m) \text{ violates a precedence or a capacity constraint} \\ &= \min(\min_{i=1, \dots, m} \mu_{[\tilde{rd}(i), +\infty)}(st_i), \min_{i=1, \dots, m} \mu_{(-\infty, \tilde{dd}(i) \ominus di]}(st_i)) \text{ otherwise.} \end{aligned} \quad (3)$$

Hence 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 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 assignments violating earliest release dates or latest completion dates are not acceptable ($\text{sat} = 0$) whereas the assignments satisfying preferred release and due dates (if they exist) are the best ($\text{sat} = 1$). Otherwise, an implicit relaxation of flexible constraints is performed, achieving a trade-off between antagonistic constraints in the spirit of [3]: our framework allows the treatment of partially inconsistent problems. In fact, the satisfaction degree of the best assignment evaluates to what extent there is an assignment satisfying all the constraints. We can define the feasibility degree of the problem by:

$$\text{Cons} = \max(st_1, \dots, st_m) \text{ Sat}(st_1, \dots, st_m) \quad (4)$$

When the constraints are partially inconsistent, $0 < \text{Cons} < 1$, since no assignment can perfectly satisfy all the constraints.

Note that contrarily to scheduling problem approaches like those defined in [14][22], a high degree of satisfaction for a constraint (e. g. total satisfaction of due date for a job *j*) cannot counterbalance a low degree for another constraint (e. g. almost violation of latest acceptable completion date for another job). Satisfaction degrees cannot be interpreted in terms of costs, but in terms of safety ranges. Indeed, it can be shown that:

$$\text{Sat}(st_1, \dots, st_m) = 0 \text{ if } (st_1, \dots, st_m) \text{ violates a capacity or a precedence constraint} \quad (5)$$

where st_j and ej_j are respectively the starting and the ending dates of job *j*. This means that $\text{Sat}(st_1, \dots, st_m)$ represents the minimal fraction of the flexibility ranges $dd_{sup}(j) - dd_{inf}(j)$ (resp.

$rd_{sup}(i) - rd_{inf}(i)$ which are left from the completion times (resp. starting times) of the jobs to 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, our approach is looking for a temporally safe schedule rather than for a low-cost schedule.

Flexible durations.

This framework can also be extended to problems with operations involving fuzzy durations, depending on their interpretation. First, durations can be subject to preferences, when they are decision variables under our control. This duration 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, and more generally, possible ranges that constrain these parameter values, in which preferences exist, and that can be modelled by fuzzy numbers (see [4]). 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.

When subject to preferences, the possible durations of operation O_i may be described by a minimal duration $dur_{inf}(i)$ and a preferred duration $dur_{sup}(i)$: $\tilde{du}(i)$ is a fuzzy number, like $\tilde{rd}(i)$ and $\tilde{dd}(i)$. The duration di of each operation O_i is then a parameter whose allowed values are restricted by the flexible constraint $di \in \tilde{du}(i)$ that we have to take into account when computing the satisfaction degree of an assignment $(s_1, \dots, s_m, d_1, \dots, d_n)$:

$$Sat(s_1, \dots, s_m, d_1, \dots, d_n) = 0 \text{ if a capacity or a precedence constraint is violated.} \quad (6)$$

$$= \min_{i=1, \dots, n} (\mu_{\tilde{du}(i)}(d_i), \mu_{\tilde{rd}(i)}(s_i), \mu_{\tilde{dd}(i)}(s_i + d_i)) \text{ otherwise}$$

The idea of the solving method is to search for an assignment (s_1, \dots, s_m) of the starting times which guarantees that there exists some possible values (d_1, \dots, d_n) such as $Sat(s_1, \dots, s_m, d_1, \dots, d_n)$ is maximal and then to compute the corresponding durations. The satisfaction degree of the best total assignment which can be obtained from (s_1, \dots, s_m) is $Sat(s_1, \dots, s_m) = \sup_{(d_1, \dots, d_n)} Sat(s_1, \dots, s_m, d_1, \dots, d_n)$. Let us first establish a more handy formulation of $Sat(s_1, \dots, s_m)$.

When the s_i are fixed such that the precedence and capacity constraints may be satisfied (i.e. when we have $s_i \leq s_j$ for all $P_i > j$ and $s_i \leq s_j$ or $s_j \leq s_i$ for all $C_i < > j$), consider, for each O_i , the set Q_i of operations which must succede O_i : $Q_i = \{j, P_i > j\} \cup \{j, C_i < > j \text{ and } s_i \leq s_j\}$. Then the values of d_i must be chosen such that $\forall j \in Q_i$ $s_j \geq s_i + d_i$, hence $s_i - d_i$ must belong to $(-\infty, \min_{j \in Q_i} s_j]$. The other constraints over the possible values of d_i are $d_i \in \tilde{du}(i)$ and $s_i - d_i \in (-\infty, \tilde{dd}(i)]$. Hence:

$$\sup_{(d_1, \dots, d_n)} Sat(s_1, \dots, s_m, d_1, \dots, d_n) = \min_{i=1, \dots, n} (\mu_{\tilde{du}(i)}(s_i - d_i), \mu_{\tilde{dd}(i)}(s_i - d_i))$$

Now, $\sup_{(d_1, \dots, d_n)} \min_{i=1, \dots, n} (\mu_{\tilde{du}(i)}(s_i - d_i), \mu_{\tilde{dd}(i)}(s_i - d_i)) = \min_{i=1, \dots, n} (\mu_{\tilde{du}(i)}(s_i - d_i), \mu_{\tilde{dd}(i)}(s_i - d_i))$

According to fuzzy arithmetic, i.e. $\sup_{(d_1, \dots, d_n)} \min_{i=1, \dots, n} (\mu_{\tilde{du}(i)}(s_i - d_i), \mu_{\tilde{dd}(i)}(s_i - d_i)) = \mu_{\tilde{du} \otimes \tilde{dd}}(s_i - d_i)$, we get:

$$\begin{aligned} &= \mu_{\tilde{du} \otimes \tilde{dd}}(s_i - d_i) \cap (-\infty, \min_{j \in Q_i} s_j] \otimes \tilde{du}(i) \quad (s_i - d_i) \\ &= \min(\mu_{\tilde{du} \otimes \tilde{dd}}(s_i - d_i), \mu_{\tilde{du}(i)}(s_i - d_i)) \cap (-\infty, \min_{j \in Q_i} s_j] \otimes \tilde{du}(i) \quad (s_i - d_i) \\ &= \min(\mu_{\tilde{du} \otimes \tilde{dd}}(s_i - d_i) \otimes \tilde{du}(i), \mu_{\tilde{du}(i)}(s_i - d_i)) \cap (-\infty, \min_{j \in Q_i} s_j] \otimes \tilde{du}(i) \quad (s_i - d_i) \end{aligned}$$

Hence, we have: $Sat(s_1, \dots, s_m) = \min_{i=1, \dots, n} (\mu_{\tilde{du}(i)}(s_i - d_i), \mu_{\tilde{dd}(i)}(s_i - d_i)) \cap (-\infty, \min_{j \in Q_i} s_j] \otimes \tilde{du}(i) \quad (s_i - d_i)$

$$= \min_{i=1, \dots, n} (\mu_{\tilde{du}(i)}(s_i - d_i), \mu_{\tilde{dd}(i)}(s_i - d_i)) \cap (-\infty, \min_{j \in Q_i} s_j] \otimes \tilde{du}(i) \quad (s_i - d_i)$$

$$= 0 \text{ otherwise}$$

In other terms, the satisfaction degree of the best total assignment which can be obtained from (s_1, \dots, s_m) is:

$$Sat(s_1, \dots, s_m) = \min_{i=1, \dots, n} (\mu_{\tilde{du}(i)}(s_i - d_i), \mu_{\tilde{dd}(i)}(s_i - d_i)) \cap (-\infty, \min_{j \in Q_i} s_j] \otimes \tilde{du}(i) \quad (7b)$$

$$\begin{aligned} &\min_{i=1, \dots, n} (\mu_{\tilde{du}(i)}(s_i - d_i), \mu_{\tilde{dd}(i)}(s_i - d_i)) \cap (-\infty, \min_{j \in Q_i} s_j] \otimes \tilde{du}(i) \\ &= \min_{i=1, \dots, n} (\mu_{\tilde{du}(i)}(s_i - d_i), \mu_{\tilde{dd}(i)}(s_i - d_i)) \cap (-\infty, \min_{j \in Q_i} s_j] \otimes \tilde{du}(i) \end{aligned}$$

In formula (7b) the feasibility degree of an assignment (s_1, \dots, s_m) can actually be expressed in terms of possibility degrees, considering that \tilde{rd} (resp. \tilde{dd} , \tilde{du}) is fuzzy restricted by $\tilde{rd}(i)$ (resp. $\tilde{dd}(i)$, $\tilde{du}(i)$):

$$\begin{aligned} Sat(s_1, \dots, s_m) &= \min_{i=1, \dots, n} (\mu_{\tilde{du}(i)}(s_i - d_i), \mu_{\tilde{dd}(i)}(s_i - d_i)) \cap (-\infty, \min_{j \in Q_i} s_j] \otimes \tilde{du}(i) \\ &= \min_{i=1, \dots, n} (\mu_{\tilde{du}(i)}(s_i - d_i), \mu_{\tilde{dd}(i)}(s_i - d_i)) \cap (-\infty, \min_{j \in Q_i} s_j] \otimes \tilde{du}(i) \\ &= \min_{i=1, \dots, n} (\mu_{\tilde{du}(i)}(s_i - d_i), \mu_{\tilde{dd}(i)}(s_i - d_i)) \cap (-\infty, \min_{j \in Q_i} s_j] \otimes \tilde{du}(i) \end{aligned} \quad (8)$$

Hence, we can first search for an assignment of the starting times such that $Sat(s_1, \dots, s_m)$ is maximal. If needed, the set of possible durations $\tilde{du}(i)$ for each task O_i corresponding to a prescribed assignment (s_1, \dots, s_m) can be then computed. The best total assignment which can be obtained from (s_1, \dots, s_m) assigns to each d_i the value with the highest degree in $\tilde{du}(i)$: $\tilde{du}(i) = \tilde{du}(i) \cap [-\infty, \tilde{dd}(i) \otimes \tilde{rd}(i)] \cap [0, \min_{j \in Q_i} (s_j - s_i)]$

$$\cap [0, \min_{j \in Q_i} \max(s_j - s_i, s_i - s_j)] \quad (9)$$

Imprecise durations.

In a second interpretation, fuzzy durations can be ill-known parameters, due to possible perturbations and can be represented by possibility distributions. In particular, if the knowledge about the durations is actually imprecise (e.g. "O_i will have a duration of approximately 5 time units"), the possibility distribution π_{d_i} describing the more or less possible values for the duration d_i of a task O_i corresponds to a trapezoidal fuzzy number $\tilde{du}(i)$ represented by the 4-tuple $(\underline{du} \leq \underline{du}^* \leq \underline{du} \leq \underline{du})$ such that $\Pi(\alpha = d_i) = \pi_{d_i}(x) = \mu_{\tilde{du}(i)}(x)$. In this case, we are looking for a schedule as robust as possible given the imprecision over the durations and the flexibility of temporal constraints. The most robust schedules are those such that all the constraints are satisfied whatever the durations will be.

A precedence constraint $P_i > j$ will be satisfied if whatever the real value of d_i , we have $s_i - s_j \geq d_i$, i.e. if $s_j - s_i$ is greater than all the possible values of d_i . In other terms, the satisfaction degree of the precedence constraint is the necessity of the fuzzy event $d_i \leq s_j - s_i$: $N(d_i \in (-\infty, s_j - s_i]) = \inf_{d_i} \max(1 - \mu_{\tilde{du}(i)}(d_i), \mu_{\tilde{du}(i)}(s_j - s_i)) = \mu_{\tilde{du}(i)}(s_j - s_i)$ (10) where $\tilde{du}(i)$ is the set of numbers necessarily after $\tilde{du}(i)$.

Similarly, the satisfaction degree of a capacity constraint $C_i < > j$ is $N(d_i \in (-\infty, s_j - s_i] \text{ or } d_j \in (-\infty, s_i - s_j])$. Since the durations are non-interactive ($\pi_{d_i d_j}(x, y) = \min(\mu_{d_i}(x), \mu_{d_j}(y))$), we have this satisfaction degree defined by:

$$\begin{aligned} N(d_i \in (-\infty, s_j - s_i] \text{ or } d_j \in (-\infty, s_i - s_j]) &= \max(N(d_i \in (-\infty, s_j - s_i]), N(d_j \in (-\infty, s_i - s_j])) \quad (11) \end{aligned}$$

Let us now study the satisfaction of a due date constraint. Since we do not know the true duration of O_i exactly, the possible values of its ending $st_i - di$ are described by a possibility distribution $\mu_{st_i \ominus \tilde{d}u(i)}(st_i - di)$, \ominus denoting the addition of fuzzy quantities. What is thus requested is that there is a possible value for dd_i , whose values are restricted by $\tilde{d}d(i)$, greater than *all* possible values of $st_i - di$. The satisfaction degree of the due date constraint is the inclusion degree of the fuzzy set $st_i \ominus \tilde{d}u(i)$ in $(-\infty, \tilde{d}d(i)]$ (instead of an intersection degree in case of flexible durations), in other terms the necessity of the fuzzy event $st_i - di \in (-\infty, \tilde{d}d(i)]$:

$$N(st_i - di \in (-\infty, \tilde{d}d(i)]) = \inf_{\tilde{d}i} \max(1 - \mu_{\tilde{d}u(i)}(di), \mu_{-\infty, \tilde{d}d(i)}(st_i - di)) \quad (12)$$

instead of its possibility degree $\Pi(\tilde{d}d(i) - di \in [st_i, +\infty)) = \Pi(st_i - di \in (-\infty, \tilde{d}d(i)])$ (see eq. (8)).

In summary, the satisfaction degree of an assignment (st_1, \dots, st_n) is defined by:

$$Sats1, \dots, st_n = \min(\min_{i=1, \dots, n} \Pi(\tilde{r}d(i) \in (-\infty, st_i]), \min_{i=1, \dots, n} N(di \in (-\infty, \tilde{d}d(i) \oplus st_i]), \min_{P_i > j} N(di \in (-\infty, st_j - st_i]), \min_{C_i < > j} N(di \in (-\infty, st_j - st_i]) \text{ or } di \in (-\infty, st_i - st_j])). \quad (13)$$

Hence, when fuzzy durations represent imprecise knowledge about the duration of a task (which is a *non controllable* parameter), 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 di is a controllable decision variable: the satisfaction degrees of due dates, precedence and capacity constraints are compatibility degrees. It is then enough that there exist durations satisfying the constraints, hence the feasibility is expressed in terms of possibility only.

3. A common framework to handle 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 formula 8) can be expressed by:

- precedence constraints $P_i -> j$: $st_j - st_i \in [\tilde{d}u(i), +\infty)$
- capacity constraints $C_i < > j$: $st_j - st_i \in [\tilde{d}u(i), +\infty)$ or $st_i - st_j \in [\tilde{d}u(j), +\infty)$
- release and due dates: $st_i \in [\tilde{r}d(i), \tilde{d}d(i) \oplus \tilde{d}u(i)]$

The scheduling problem defines a non conjunctive graph whose nodes represent the operations. Precedence and temporal constraints define the conjunctive part of the graph: a fuzzy temporal window $[\tilde{r}d(i), \tilde{d}d(i)]$ is associated to each node. Each conjunctive edge $P_i -> j$ represents a precedence constraint "O_i must precede O_j", by means of a fuzzy inequality of the type $st_j - st_i \geq \tilde{d}u(i)$. Capacity constraints $C_i < > j$ define non conjunctive edges ($st_j - st_i \geq \tilde{d}u(i)$ OR $st_i - st_j \geq \tilde{d}u(j)$) which represent conflicts of the type "O_i before O_j OR O_j before O_i".

This kind of graph can also represent constraints involving ill-known durations. Indeed, in this case, a precedence constraint $P_i -> j$ requires that $st_j - st_i$ belongs to $[\tilde{d}u(i), +\infty)$ the set of values which are *necessarily* greater than $\tilde{d}u(i)$, or, in other terms, that $st_j - st_i$ belongs to $[\tilde{d}u(i), +\infty)$, the set of values that are possibly greater than $\tilde{d}u(i)$, $\tilde{d}u(i)$ being defined by $\tilde{d}u(i) = \inf_{\tilde{d}u(i)} * \sup_{\tilde{d}u(i)}$ (see fig. 3).

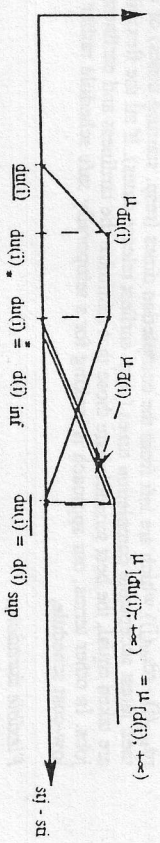


Figure 3: from $[\tilde{d}u(i), +\infty)$ to $[\tilde{d}u(i), +\infty)$

A due date constraint requires that $st_i \ominus \tilde{d}u(i)$ is included in the core of $(-\infty, \tilde{d}d(i)]$, or equivalently that st_i belongs to $(-\infty, \tilde{d}d(i) \oplus \tilde{d}u(i)]$. Hence, the different types of constraints over the possible values of the starting times can be expressed by:

- precedence constraints $P_i -> j$: $st_j - st_i \in [\tilde{d}u(i), +\infty)$
- capacity constraints $C_i < > j$: $st_j - st_i \in [\tilde{d}u(i), +\infty)$ or $st_i - st_j \in [\tilde{d}u(j), +\infty)$
- release and due dates: $st_i \in [\tilde{r}d(i), \tilde{d}d(i) \oplus \tilde{d}u(i)]$

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 duration parameters over the graph come from the specification of preferences and represent the possible values that can be assigned to the di 's. In case of imprecisely known durations, these parameters comes from the uncertainty about the real value of some durations: each of them represents the set of values that are necessarily greater than the duration of an operation. In the following, we suppose that fuzzy durations $\tilde{d}u(i)$ represent flexible durations, knowing that imprecise durations can be handled, replacing $\tilde{d}u(i)$ by $\tilde{d}u(i)$.

4. Solving paradigms.

Our solving approach consists in searching for a sequencing of operations (like in [1][6][14]), from which earliest and latest corresponding schedules can be computed. In other terms, we have to transform disjunctive constraints "O_i before O_j OR O_j before O_i" into conjunctive ones by choosing one of the alternatives. It is based on three basic procedures:

- a general search procedure,
- a consistency enforcing procedure which propagates decisions through the network,
- a constraint analysis procedure (or look ahead analysis) that determines which decision to make next.

Consistency Enforcing.

Considering only the conjunctive part of the graph, each precedence constraint $P_i -> j$ implies that the fuzzy temporal window associated to O_j (resp. O_i) must be such that $rj \geq rdi + di$ (resp. $ddi \leq ddi - dj$). Hence, the temporal windows can be updated as follows:

$$\begin{aligned} rdn(i) &:= \max(rdn(i), rdn(i) + \tilde{d}u(n(i))) & rdsup(j) &:= \max(rdsup(j), rdsup(j) + \tilde{d}u(n(i))) \\ ddn(i) &:= \min(ddn(i), ddn(i) - \tilde{d}u(n(i))) & ddsup(j) &:= \min(ddsup(j), ddsup(j) - \tilde{d}u(n(i))) \end{aligned}$$

A linear algorithm has been implemented which updates the nodes (i. e. the fuzzy temporal windows) according to the precedence constraints. This algorithm is an adaptation of classical shortest or longest path algorithms to fuzzy networks (see e. g. [5]) that takes advantage of the acyclicity of the graph to produce an efficient ordering for updating the temporal windows.

It turns out that this method guarantees that the best of the earliest (resp. latest) schedules according to the precedence constraints can then be obtained when assigning to each st_i the lowest (resp. greatest) date among its best possible values, i. e. values with highest membership degree in the set $[\tilde{r}d(i), \tilde{d}d(i) - \tilde{d}u(i)]$.

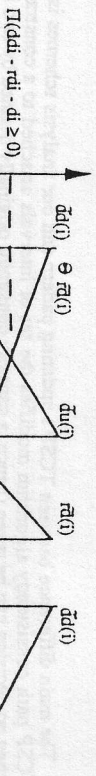


Figure 4: consistency degree of O_j 's temporal window.

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:

$$\begin{aligned} \text{Cons(conjunctive part)} &= \min(\min_{i=1..n} \mu_{\tilde{r}(i)}(\text{st}), \min_{i=1..n} \mu_{\tilde{d}(i)}(\text{st})) \\ &= \min_{i=1..n} \Pi(d(i) - r(i) - dt \geq 0) \end{aligned} \quad (14)$$

Cons(conjunctive part) considering 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.

Search Procedure

The sequencing we are searching for is one of those having the best satisfaction degree. The search procedure is in fact a classical branch&bound algorithm using a depth-first strategy. The nodes of the tree represent partial sequencing and its leaves complete sequencing: extending a node means choosing a disjunction (O_i precedes O_j OR O_j precedes O_i) and 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 $\text{dd}(i)$ backwards and $\tilde{r}(i)$ forwards through the new edge. Hence we get an estimate of Cons(conjunctive part) and associate it 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 extended. Otherwise, the algorithm backtracks to a node whose degree is greater than 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 [15]) based on the same principles and integrating different variants (see [23]).

Look ahead procedure

The efficiency of the search relies of course on the search heuristic that determines which disjunction to be instantiated next. It is actually based on the notion of constraint analysis [10][11]. For each disjunction (O_i precedes O_j OR O_j precedes O_i), an upper bound of the possibility of each alternative may be computed [5]:

$$\begin{aligned} \Pi_{\text{sup}}(O_i \text{ precedes } O_j) &= \Pi(d(j) - r(i) - (dt + dj) \geq 0) \\ &= \sup_{x \geq 0} \mu_{\tilde{d}(j)}(\alpha \oplus \tilde{r}(i) \oplus \tilde{d}(j)) \otimes \mu_{\tilde{r}(i)}(\alpha \oplus \tilde{d}(j)) \otimes \mu_{\tilde{d}(j)}(x) \\ &= \mu_{\tilde{d}(j)}(\alpha \oplus \tilde{r}(i) \oplus \tilde{d}(j)) \otimes \mu_{\tilde{r}(i)}(\alpha \oplus \tilde{d}(j)) \end{aligned} \quad (15)$$

Note that $\max(\Pi_{\text{sup}}(O_i \text{ precedes } O_j), \Pi_{\text{sup}}(O_j \text{ precedes } O_i))$ 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 as $\max(\Pi_{\text{sup}}(O_i \text{ precedes } O_j), \Pi_{\text{sup}}(O_j \text{ precedes } O_i)) \leq \alpha$, α being the satisfaction degree of the current best instantiation.

$\Pi_{\text{sup}}(O_i \text{ precedes } O_j) = 0$ means that the decision " O_i precedes O_j " is inconsistent in the current search state according to O_i and O_j 's temporal windows. Hence, decision " O_j precedes O_i " must be performed (otherwise, the satisfaction degree of the sequencing will be 0). In fact, $\min(\Pi_{\text{sup}}(O_i \text{ precedes } O_j), \Pi_{\text{sup}}(O_j \text{ precedes } O_i))$ estimates the degree to which the satisfaction will decrease if the best of the two alternatives is not chosen. The necessity to choose the best alternative, in other terms, the cronicity of the conflict, is defined by:

$$C(O_i, O_j) = 1 - \min(\Pi_{\text{sup}}(O_i \text{ precedes } O_j), \Pi_{\text{sup}}(O_j \text{ precedes } O_i)) \quad (16)$$

The cronicity of a resource is the maximal cronicity of the conflicts between operations to which the resource has been assigned.

Hence, the look ahead procedure first computes the cronicity of each conflict. Note that 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 consistency enforcing procedure while creating the search state (the cronicity of the others remain unchanged). In order to keep the satisfaction degree as high as possible, a conflict whose cronicity is maximal is then chosen to be instantiated by the most possible alternative (alternatives (O_i precedes O_j) such as $\Pi_{\text{sup}}(O_i \text{ precedes } O_j) \leq \alpha$ will lead to a solution worse than the best current one: they do not have to be extended). If several conflicts correspond to the maximal cronicity, one can focus on the machine having the larger set of critical conflicts. Since the set of the critical conflicts happening on a machine is a fuzzy set, this cardinality is actually a fuzzy cardinality.

Although our framework is not additive like the one proposed in [22], our heuristic is similar: the lower the quality of the best scheduling deriving from an alternative, 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 [16], in which the most possible decisions are chosen regardless of their degrees of necessities. In that work, the priority of a conflict is the maximum possibility of the alternative: a conflict involving two equally possible alternatives is paradoxically considered as interesting as a conflict involving one impossible alternative and one alternative as possible as the previous one.

This kind of procedure can be coupled with a knowledge-based decision support module (see for instance [1][7]): 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 (the highest cronicity is lower than $1 - \text{Cons(conjunctive part of the current node)}$). It is then better to use other choice criteria, especially knowledge-based criteria (which can also use the possibility degrees computed by the constraint analysis procedure).

Note that the calculation $\Pi_{\text{sup}}(O_i \text{ precedes } O_j)$ according to (15) gives only an upper bound of $\Pi(O_i \text{ precedes } O_j)$. For instance, consider a crisp case involving three tasks conflicting for the same resource: $\text{du}(i) = \text{du}(j) = \text{du}(k) = 2$, $\text{rd}(j) = 0$, $\text{rd}(i) = \text{rd}(k) = 1$ and $\text{dd}(i) = \text{dd}(j) = \text{dd}(k) = 6$. We obtain $\Pi_{\text{sup}}(O_i \text{ precedes } O_j) = \Pi_{\text{sup}}(O_j \text{ precedes } O_i) = 1$ although no scheduling placing i before j is feasible.

More elaborate analyses of the conflicts can be performed, taking more than two operations into account. Constraint analysis rules like those proposed in [11] can be easily extended to the fuzzy case. The difficulty of this method is that it generates capacity constraints involving more than two tasks (e.g. " O_i must precedes O_j or O_k "), even in no fuzzy problems. These new obtained constraint are not easy to use in a look ahead strategy.

We propose new analysis rules involving more than two nodes which do not present this drawback. For instance, an approximation of $\Pi_{\text{sup}}(O_i \text{ precedes } O_j)$ better than the one

computed by (16) can be obtained taking three operations into account, say O_i , O_j and O_k , and letting Ok vary:

$$\Pi_{sup}(O_i \text{ precedes } O_j) = \min (\Pi((ddj - rdi) \cdot (di + dj) \geq 0),$$

$$\min Ok (\max \Pi((ddj - rdi) \cdot (di + dj + dk) \geq 0),$$

$$\Pi((ddj - rdk) \cdot (di + dj + dk) \geq 0),$$

$$\Pi((ddk - rdi) \cdot (di + dj + dk) \geq 0))$$

where the three additional terms pertain to the respective sequences: $O_i/O_k/O_j$, $O_k/O_i/O_j$ and $O_i/O_j/O_k$. Coming back to our example, we obtain $\Pi_{sup}(O_i \text{ precedes } O_j) = 0$, which enforces the other precedence constraint.

5. Scheduling problems as Constraint Satisfaction Problems (CSPs).

As outlined in [22], scheduling problems can be understood as particular Constraint Satisfaction Problems (CSP) [18][20]. In this section, we emphasize this comparison, in order to relate each of our solving procedures (reduced to the crisp case) with existing CSP solving procedures.

R. Dechter, I. Meiri and J. Pearl [2] define Temporal Constraint Satisfaction Problems (TCSPs) as binary Constraint Satisfaction Problems (CSPs) whose variables take real values. In TCSP, every constraint T_{ij} relating a pair of variables (x_i, x_j) is defined by a set of intervals $\{I_{ij}^1, \dots, I_{ij}^n\}$ meaning that $x_i - x_j$ must belong to $I_{ij}^1 \cup \dots \cup I_{ij}^n$. 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 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, $st0$ 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 st_i to $st0$: $st_i - st0 \in [rd(i), dd(i) - du(i)]$
- precedence constraints $P_i \rightarrow j$: $st_j - st_i \in [du(i), +\infty)$
- capacity constraints $C_i < > j$: $st_j - st_i \in [du(i), +\infty) \cup (-\infty, -du(j)]$

Enforcing arc consistency in a TCSP consists of iteratively considering each pair of variables (x_i, x_j) related by a constraint T_{ij} (resp. T_{ji}) in order to apply the updating pattern $T_{ii} := T_{ii} \cap (T_{ij} - I_{ij}^1 \cup \dots \cup T_{ij} - I_{ij}^n)$ (resp. $T_{ii} := T_{ii} \cap (T_{ji} + I_{ji}^1 \cup \dots \cup T_{ji} + I_{ji}^n)$). If T_{ij} represents a precedence constraint, these updating patterns correspond respectively to backward and forward propagation of the starting times. In this context, the consistency enforcing procedure is nothing but an improved arc-consistency procedure, limited to the conjunctive part of the TCSP: it ensures that the set of possible starting times of each operation is consistent with the possible starting times of each of its neighbours.

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} et T_{kj} in order to apply the updating pattern $T_{ij} := T_{ij} \cap (\cup_{k \in N} I_{ka} + I_{kb})$. When $T_{ij} = C_i < > j$ and x_k stands for the starting time of an operation Ok , the updating pattern ensures that the constraint between O_i and O_j is coherent with Ok : it corresponds to our second analysis scheme (17). When $x_k = st0$, and $T_{ij} = C_i < > j$, this updating pattern corresponds to our first constraint analysis scheme (15). Indeed,

$$[-dd(i) + du(i), -rd(i)] \cup [rd(j), dd(j) - du(j)] = [rd(j) - dd(i) + du(i), dd(j) - du(j) - rd(i)]$$

and:

$$[du(i), +\infty) \cup (-\infty, -du(j)) \cap [rd(j) - dd(i) + du(i), dd(j) - du(j) - rd(i)]$$

$$= [\max(du(i), rd(j) - dd(i) + du(i)), dd(j) - du(j) - rd(i)] \text{ (decision } O_i \text{ before } O_j)$$

$$= [rd(j) - dd(i) + du(i), \min(-du(j), dd(j) - du(j) - rd(i))] \text{ (decision } O_j \text{ before } O_i)$$

$$\text{when } du(j) + du(i) \leq dd(j) - rd(i) \text{ and } du(j) + du(i) > dd(i) - rd(i)$$

$$\text{when } du(j) + du(i) > dd(j) - rd(i) \text{ and } du(j) + du(i) \leq dd(i) - rd(i)$$

The main difference between TCSP's updating pattern and our analysis schemes 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 intervals can be transformed into a constraint involving more intervals). This modification is then propagated to others 3-tuples of tasks. In constraint analysis, we 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.

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_{ij}). Once a decision has been taken, it is propagated (i. e. consistency enforcing and constraint analysis procedures are applied) as it is the case in classical CSP when a look ahead procedure is used.

Hence, when reduced to the crisp case, scheduling problems are particular TCSPs, which are themselves particular CSPs. Moreover, our solving procedures can be understood as specialized (and thus improved) versions of some classical CSP algorithms. This remark suggests to adapt 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 [22] has shown that translating without modification CSP heuristics to the scheduling domain reveals inefficient. A more promising research area for both domains is the study of flexible problems. For instance, the study of scheduling problems involving costs [22] may define another kind of additive CSPs than those defined by Freuder [13]. Moreover, fuzzy scheduling problems can be viewed as dedicated Fuzzy Constraint Satisfaction Problems [6].

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. consistency enforcing in a conjunctive graph) and constraint analysis schemes can be easily extended to this new framework.

Experiments were made [19] to compare fuzzy constraint analysis (based on (16)) involving flexible release dates and due dates to crisp constraint analysis in the framework of the Opal system [1] (the first version of Opal only assumed crisp windows corresponding to the earliest release date and the latest acceptable completion date). Except for strongly constrained problems where a crisp analysis can make most of the decisions without referring to the knowledge-based module, these experiments show that fuzzy analysis is more productive than crisp analysis. The integration of fuzzy durations has also been implemented and its computational comparison to crisp durations over a real size problem will be soon performed. Moreover, the more elaborate look-ahead scheme (19) is currently under experiments to determine whether it really enhances both the predictiveness of the search procedure and 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).

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 change significantly 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, we avoid empirical relaxations techniques which often happen to be more expensive, difficult to formulate, and suboptimal. Moreover, this framework can be easily extended to capture priority between constraints (e. g. "it is preferable to schedule O_i before O_j ") see [6]. It should also be noticed that the fuzzy approach may handle partially inconsistent problems. A solution (the instantiation with the maximal satisfaction degree) will be provided as long as the problem is not totally inconsistent. Finally, we must outline that our formalism suggests a non-monotonic framework for dynamic scheduling problems as for dynamic fuzzy Constraint Satisfaction Problems [6].

In this paper, we have presented an approach for computing predictive schedules taking into account flexibilities and uncertainty. Clearly, we may also think of taking advantage of fuzzy constraints in a real-time monitoring perspective.

REFERENCES

- [1] E. Bensana, G. Bel, D. Dubois, 1988, "OPAL, a multi-knowledge-based system for industrial job-shop scheduling" *International Journal of Production Research*, 26(5), 795-816.
- [2] R. Dechter, I. Meiri, J. Pearl, 1991, "Temporal Constraint Networks". *Artificial Intelligence* 49, 61-95.
- [3] Y. Dessoues, J.C. Latombe, 1985, "Making compromises among antagonist constraints". *Artificial Intelligence* 27, 159-174.
- [4] D. Dubois, 1987, "An Application of Fuzzy Arithmetics to the Optimization of Industrial Machining Processes". *Mathematical Modelling*, 9, 461-475.
- [5] D. Dubois, 1989, "Fuzzy knowledge in an artificial intelligence system for job-shop scheduling" in *Applications of the Fuzzy Set Methodologies in Industrial Engineering*, G. Evans, W. Karwowski and M. Wilhelm Eds, Elsevier, 73-89.
- [6] D. Dubois, H. Fargier, H. Prade, 1993, "Propagation and satisfaction of flexible constraints", in *Fuzzy Sets, Neural Networks and Soft Computing*, R. R. Yager and L. A. Zadeh eds, Kluwer Academic, Publ.
- [7] D. Dubois, J. L. Koling, H. Prade, 1991, "DEBORCA, a decision engine based on Rational Aggregation", *2nd Annual Conf. on A.I. Simulation and Planning in High Autonomy Systems*, Cocoa Beach (FL), IEEE Comp. Soc. Press, 68-77.
- [8] D. Dubois, H. Prade, 1988, with the collaboration of H. Fargery, R. Martin-Clochaire and C. Testemale *Possibility Theory - An Approach to Computerized Processing of Uncertainty*, NY: Plenum Press.
- [9] D. Dubois, H. Prade, 1989, "Processing fuzzy temporal knowledge", *IEEE Trans. on System, Man and Cybernetics*, 19(4), 729-744.
- [10] J. Ershler, 1976, "Finding some essential characteristics of the feasible solutions for a scheduling problem". *Operations Research*, 24, 774-783.
- [11] J. Ershler, P. Esquivel, 1986, "Decision-aid in job shop scheduling: A knowledge based approach", *Proceedings of the 1986 IEEE Conference on Robotics and Automation, San Francisco*, 1751-1756.
- [12] J. Ershler, P. Lopez, C. Thuniel, 1989, "Temporal reasoning under resources constraints: application to task scheduling", *2nd Int. Symposium on systems Research, Informatics and Cybernetics*, Baden-Baden.
- [13] Freuder E. C., 1992, *Partial Constraint Satisfaction*, AI 58, nov 92, pp 24-71.
- [14] M. S. Fox, S. F. Smith, 1984, "SIS: a knowledge based system for factory scheduling", *Expert Systems* 1(1), 25-49.
- [15] M. S. Fox, G. A. Strohm, 1982, "Job-shop scheduling: An investigation in constraint-directed reasoning", *Proceedings, National Conference on Artificial Intelligence*, Pittsburgh.
- [16] R. M. Kerr, and R. N. Walker, 1989, "A job shop scheduling system based on fuzzy arithmetics", *Proceedings 3rd Int. Conf. on Expert Systems and the Leading Edge in Production and Operation Management*, Hilton Head, SC 453-450.
- [17] C. Lepepe, 1985, "SOJA: a daily workshop scheduling system", *Proceedings ECS Specialist Group on Expert Systems Conference*, Warwick, 225-242.
- [18] Mackworth A. K., 1977, "Consistency in networks of relations", *Artificial Intelligence* 8, 99-121.
- [19] N. Mabbé, 1987, "Prise en compte de l'imprécision des délais dans la construction d'ordonnements prévisionnels", Master Degree Thesis, Univ. P. Sabatier, Toulouse - in French.
- [20] Montanan H., "Networks of Constraints: Fundamental Properties and Application to Picture Processing", *Information Science* 7, 1974, pp 95-142.
- [21] P. S. Ow, S. F. Smith, 1986, "Toward an opportunistic scheduling system", *Proc. 19th Hawaii International Conference on Systems Science*.
- [22] N. Sadeh, 1991, "Look-ahead techniques for micro-opportunistic job shop scheduling", Report CSS1-102, Carnegie Mellon Univ., Pittsburgh.
- [23] T. Schiec, 1992, "Possibilistic constraint satisfaction problems or how to handle soft constraints", *Proceedings of the 8th Conference on Uncertainty in Artificial Intelligence*, Stanford, July 19-22, 268-275.
- [24] L.A. Zadeh, 1978, "Fuzzy sets as a basis for a theory of possibility", *Fuzzy Sets & Systems*, 1, 3-28.