ELSEVIER

European Journal of Operational Research 118 (1999) 95–126

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/orms

Theory and Methodology

Computing improved optimal solutions to max–min flexible constraint satisfaction problems

Didier Dubois [a], Philippe Fortemps [b,*]

[a] Université Paul Sabatier, 118 route de Narbonne, 31062 Toulouse Cedex, France

[b] Faculté Polytechnique de Mons, Department of Mathematics and Operational Research, 9 rue de Houdain, B-7000 Mons, Belgium

Received 10 September 1997; accepted 30 June 1998

Abstract

The formal framework for decision making in a fuzzy environment is based on a general max–min, bottleneck-like optimization problem, proposed by Zadeh. It is also the basis for extending the constraint satisfaction paradigm of Artificial Intelligence to accommodating flexible or prioritized constraints. This paper surveys refinements of the ordering of solutions supplied by the max–min formulation, namely the discrimin partial ordering and the leximin complete preordering. A general algorithm is given which computes all maximal solutions in the sense of these relations. It also sheds light on the structure of the set of best solutions. Moreover, classes of problems for which there is a unique best discrimin and leximin solution are exhibited, namely, continuous problems with convex domains, and so called isotonic problems. Noticeable examples of such problems are fuzzy linear programming problems and fuzzy PERT-like scheduling problems. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Fuzzy sets; Constraint satisfaction problem; Flexible constraint; Flexible mathematical programming; Leximin order

1. Introduction

Flexible constraint satisfaction problems (FCSP) come from putting together the fuzzy approach of Bellman and Zadeh (1970) to multiple criteria decision-making and the constraint-directed methodology for combinatorial problem-solving, popular in the field of Artificial Intelligence (Van Hentenryck, 1989). Constraint Satisfaction is a generic framework for combinatorial problems whose aim is to find one or all solutions to a set of constraints. Artificial Intelligence offers efficient tools for representing generic constraints in an extensional way, and constraint-based programming languages handle such constraints in an intensional way (Esquirol et al., 1995). Independently, and well before the constraint-directed problem-solving paradigm came to light, Bellman and Zadeh revisited the setting of multiple criteria

[*] Corresponding author. E-mail: philippe.fortemps@fpms.ac.be.

PII: S 0 3 7 7 - 2 2 1 7 (9 8) 0 0 3 0 7 - 5

optimization and proposed fuzzy sets as a unique tool for representing constraints and criteria. A fuzzy subset over a set of potential solutions to a problem represents "good" solutions in some sense and $\mu_{F_i}(s)$ represents the degree of attainment of goal i by solution s, or alternatively the degree of satisfaction of constraint i by solution s. An optimal solution to a set $\{F_1, \dots, F_n\}$ of criteria or constraints is defined as any s^* such that $\min(\mu_{F_1}(s^*), \dots, \mu_{F_n}(s^*))$ is maximal. The consistency between this approach and the constraint satisfaction paradigm of artificial intelligence is obvious if the fuzzy model is particularized to the case where F_1, …, F_n are crisp subsets (or relations): then finding a max–min optimal solution comes down to finding a feasible solution. As a consequence, the fuzzy sets in the Bellman–Zadeh approach are more convincingly interpreted as flexible constraints rather than objective functions. Indeed the lack of compensation between F_i, due to the minimum operation, makes full sense in terms of constraints: one fully violated constraint is enough to make the problem infeasible, while multiple criteria-decision-making often involves compensatory connectives (e.g. Keeney and Raiffa, 1976).

The maximin framework for constraint-satisfaction problems has been first suggested by Freuder and Snow (1989) and fully developed by Dubois, Fargier and Prade (Dubois et al., 1996; Dubois et al., 1994; Fargier, 1994). A major advantage of this framework is that not only it enables a better discrimination between good and less good solutions to a set of constraints, but all the machinery of constraint propagation extends over to flexible constraints: levels of preference can be propagated using simple extensions of existing algorithms (e.g. Fargier, 1994; Dubois et al., 1996), so as to visualize fuzzy sets of feasible solutions to a problem. This approach has been successfully applied to jobshop scheduling (Dubois et al., 1995; Fargier, 1997) where the fuzzy constraint formulation is general enough to subsume constraint based approaches (Erschler et al., 1976; Erschler et al., 1991) as well as some single criterion optimization methods.

However the maximin approach is limited by a lack of discrimination among the solutions to a flexible constraint satisfaction problem. Namely two solutions whose levels of satisfaction of the most violated constraint are identical will not be told apart even if they could be distinguished by considering levels of satisfaction of other constraints. In order to remedy this situation, two refinements to the min-based ordering of solutions have been proposed (Fargier et al., 1993) and characterized (Dubois et al., 1995). They are related respectively to a partial ordering (we call "discrimin") of preferred subsets of logical formulas in an inconsistent knowledge base as proposed by Brewka (1989), and to a social welfare ranking function called "leximin" (Moulin, 1988; Sen, 1986). However, the question of how to compute such improved optimal solutions to flexible CSP's remains open. The present paper is a preliminary investigation into this question. We define a general algorithm to compute either the discrimin or the leximin optimal solutions. Thus classes of problems are derived for which the discrimin optimal solution is unique. These classes of problems contain as particular cases project scheduling problems, traveling salesman problems, and special forms of fuzzy linear constraint problems.

The paper is organized as follows: first, the framework of constraint satisfaction problems is recalled, and its extension to flexible constraints inside the Bellman–Zadeh model as well. Several refinements of the min-based ordering are presented. Then Section 3 exemplifies the limitation of the min-based ordering of solutions on the project scheduling problem with flexible ready-dates, flexible due-dates and flexible activity durations modelled by fuzzy sets. A procedure that computes the leximin-optimal solution to this particular problem is suggested. Section 4 presents the general algorithm, as well as some direct modifications of it to particular combinatorial problems or to constraint satisfaction problems with fuzzy domains. Section 5 gives an illustration of the proposed algorithm on a bottleneck assignment problem. Section 6 starts the study of some important particular cases of the problem, namely convex or isotonic problems. For the classes of convex or isotonic problems, we prove the uniqueness of the discrimin solution.

2. Flexible constraint satisfaction problems

2.1. Constraint satisfaction problems

Constraint satisfaction defines a general framework for decision problems (Mackworth, 1977; Montanari, 1974; Sadeh, 1991; Van Hentenryck, 1989). Solving a constraint satisfaction problem is achieved by means of a depth-first technique that tries to construct a solution via a progressive instantiation of variables, and "intelligent" backtrack procedures that can undo previous decisions that were proved infeasible. On the other hand, a constraint propagation is often used to compute the effects of one decision on the set of uninstanciated variables and thus reduce the set of admissible values of all these variables. The combined use of these two techniques allows to efficiently determine if the problem is feasible, and can as well generate all its feasible solutions.

The set of all potential solutions (feasible or not) Ω is the Cartesian product of the variables definition domains. Each choice of values for those variables is called "a solution". A relation [1] R represents the solutions $\bar{d}$ satisfying the constraint C:

$$\bar{d} \in R \quad \text{means the constraint is satisfied,} \tag{1}$$

$$\bar{d} \notin R \quad \text{means the constraint is violated.} \tag{2}$$

More formally, a CSP $\mathscr{P} = (\mathscr{X}, \mathscr{D}, \mathscr{C}, \mathscr{R})$ is defined by

- a set of decision variables $\mathscr{X} = \{X_1, \ldots, X_n\}$;
- a set of domains $\mathscr{D} = \{D_1, \ldots, D_n\}$ where D_i is the domain of X_i. Ω, the solutions set, is the Cartesian product of the domains: $\Omega = D_1 \times \cdots \times D_n$;
- a set of constraints $\mathscr{C} = \{C_1, \ldots, C_m\}$;
- a set of binary relations $\mathscr{R} = \{R_1, \ldots, R_m\}$ where R_j defines the set of solutions satisfying the constraint C_j.

The *feasible solution* set, $\text{Sols}(\mathscr{P})$, is defined by the satisfaction of all constraints. Therefore, it is the intersection of the different satisfaction sets R_i for each constraint C_i:

$$\text{Sols}(\mathscr{P}) = R_1 \cap \cdots \cap R_m. \tag{3}$$

As a matter of fact, this framework does not deal with any objective function. The choice of the final solution inside $\text{Sols}(\mathscr{P})$ belongs to the Decision Maker (DM). In practice, a solution is often picked at random by constraint satisfaction software. If the solution set is non-empty, the problem is said to be *consistent* and there exists at least one feasible solution.

An important example of CSP is the project scheduling problem. In a scheduling problem, the three main kinds of constraints are the following ones (Erschler et al., 1976; Erschler et al., 1991):

- *Precedence constraints*: Typically an operation may not start before another finishes.
- *Temporal constraints*: An operation O_i has to be performed inside a temporal window, defined by a ready date RD_i and a due date DD_i.
- *Capacity constraints*: Each operation requires a certain amount of resources and these resources are in a limited quantity. For instance, in Jobshop Scheduling, a machine may be allocated to at most one operation at a time.

In the description of the constraints, we use lower cases to denote variables and upper cases for data. If the decision variable related to the operation O_i is its starting time, s_i, the precedence constraints may be written

$$s_k - s_i \geqslant DU_i \quad (O_i \text{ before } O_k), \tag{4}$$

where DU_i is the duration of operation O_i; the capacity constraints become, if O_i and O_j require the same resource,

$$s_j - s_i \geqslant DU_i \text{ or } s_i - s_j \geqslant DU_j \tag{5}$$

and the temporal constraints (ready date and due date) are

$$s_i \geqslant RD_i, \tag{6}$$

$$s_i \leqslant DD_i - DU_i, \tag{7}$$

by denoting RD_i (resp. DU_i and DD_i) the ready date (resp. the duration and the due date) of O_i.

[1] The notation R will be used for the set this relation defines in Ω: $\bar{d} \in R \iff R(\bar{d}) = 1$.

The usual technique for solving a constraint-directed Jobshop scheduling problem is to get rid of the disjunctive constraints (5) by choosing one side of each disjunction. Each problem obtained by making such a choice for all disjunctive constraints reduces to a simpler project scheduling problem where the only decisions left are the starting time of operations.

When no disjunctive constraint is left, classical PERT-CPM algorithms enable a smallest time-window to be computed for each operation i, of the form $[rd_i, dd_i - DU_i)$ where rd_i is the earliest starting time of operation i and dd_i the latest ending time. This propagation is made via precedence constraints.

The constraints then have the following form (by denoting Γ_i the set of indices of the operations preceded by O_i):

$$s_i \in [rd_i, dd_i - DU_i], \tag{8}$$

$$s_k - s_i \in [DU_i, +\infty) \quad \forall k \in \Gamma_i, \tag{9}$$

where the initial value of rd_i (resp. dd_i) is RD_i (resp. DD_i). In this way, the first relation (8) builds the initial temporal window for the variable s_i. By propagating the second constraint (9), this temporal window is reduced, until stability is reached.

If finally the temporal window of each variable s_i is non empty, there exists at least one feasible solution. Otherwise no solution is feasible. See Erschler et al. (1976) and Erschler et al. (1991) for strategies that cope with disjunctive constraints (5).

2.2. Flexible constraint satisfaction problems

The Flexible Constraint Satisfaction Problem (FCSP) (Fargier, 1994) generalizes the classical Constraint Satisfaction Problem (CSP). The latter indeed faces two difficulties: sometimes a CSP has no solution, but would have one if a non-significant change of constraints were made. In other cases, the number of feasible solutions can be very high, and the decision making is left with no guideline to choose among them. In fact, it can be useful to represent constraints involving uncertainties on the data values or the decision-maker (DM's) preferences on the solution; also it may avoid artificial cases of inconsistency due to too rigid a statement of constraints. Possibility theory (Zadeh, 1978; Dubois and Prade, 1987) allows the mathematical modeling and treatment of flexible constraints. These flexible constraints will be represented by fuzzy relations.

Typically a flexible constraint may be partially satisfied only but its full satisfaction is preferred. The main notion here is the satisfaction degree of a flexible constraint by a solution.

The set of all solutions (feasible, non-feasible or partially feasible) Ω is, as previously, the Cartesian product of the definition sets of all variables. Each choice of values for these variables is considered as a *solution*.

The degree to which a solution $\bar{d} \in \Omega$ satisfies a constraint C will be described by the value $\mu_R(\bar{d})$ – where R is the fuzzy relation related to the constraint C. For all elements $\bar{d} \in \Omega$, R gives the preference level according to the constraint C:

$$\begin{aligned}
&\mu_R\colon \Omega \to [0,1],\\
&\mu_R(\bar{d}) = 1 \quad \text{means } \bar{d} \text{ totally satisfies } C,\\
&\mu_R(\bar{d}) = 0 \quad \text{means } \bar{d} \text{ totally violates } C,\\
&\mu_R(\bar{d}) \ \in]0,1[\quad \text{means } \bar{d} \text{ partially satisfies } C.
\end{aligned}$$

A flexible constraint C can be described by *a fuzzy relation R* or by the fuzzy set defined by R in Ω. R is the fuzzy set of solutions satisfying the flexible constraint C.

Definition 1. The intersection of two fuzzy relations R_i and R_j is the fuzzy relation $R_i \cap R_j$ defined by [2]

$$\mu_{R_i \cap R_j}(\bar{d}) = \min(\mu_{R_i}(\bar{d}), \mu_{R_j}(\bar{d})).$$

Therefore, $\mu_{R_i \cap R_j}(\bar{d})$ represents to what extent the constraints C_i and C_j are simultaneously respected by $\bar{d}$.

A *FCSP* (Fargier, 1994; Dubois et al., 1994, 1996) $\mathscr{P} = (\mathscr{X}, \mathscr{D}, \mathscr{C}, \mathscr{R})$ is defined by:

[2] We assume both fuzzy relations defined on the same space. If not, we would have to use the cylindrical extensions of the relations on the union of the spaces.

- A set of variables $\mathscr{X} = \{X_1, \ldots, X_n\}$.
- A set of definition domains $\mathscr{D} = \{D_1, \ldots, D_n\}$ where D_i is the definition domain of X_i. Ω is the Cartesian product of the definition domains $\Omega = X_1 \times \cdots \times X_n$.
- A set of constraints $\mathscr{C} = \{C_1, \ldots, C_m\}$. They can be either flexible or classical.
- A set of fuzzy relations $\mathscr{R} = \{R_1, \ldots, R_m\}$ where R_j defines the solutions satisfying more or less the constraint C_j. For the classical constraints, the relation R_j is all-or-nothing, while for the flexible constraints, R_j is a fuzzy relation.

For each solution $\bar{d}$, the global satisfaction degree is

$$\mathrm{Sat}(\bar{d}) = \mu_{R_1 \cap \cdots \cap R_m}(\bar{d}) = \min_{C_i \in \mathscr{C}} \mu_{R_i}(\bar{d}). \tag{10}$$

In such an approach, a strong assumption has been made: all the preferences, represented by the fuzzy relations R_j, can be measured with the same scale. Therefore, it makes sense to use the "min" operator in order to compute the global satisfaction degree. It gives, in fact, the same weight to every constraint, even if one of them is stated several times. As a consequence it is innocuous that the fuzzy constraints be dependent, since the minimum operation is idempotent.

This egalitarian approach implies that the satisfaction degree of $\mathscr{P}$ by $\bar{d}$ is the satisfaction degree of the least satisfied constraint. Moreover, $\mathrm{Sat}(\bar{d})$ gives the degree to which $\bar{d}$ belongs to the fuzzy set of the feasible solutions of $\mathscr{P}$, $\mathrm{Sols}(\mathscr{P})$:

$$\mu_{\mathrm{Sols}(\mathscr{P})}(\bar{d}) = \mathrm{Sat}(\bar{d}).$$

Therefore, $\mathrm{Sols}(\mathscr{P}) = R_1 \cap R_2 \cap \cdots \cap R_m$ still holds using the fuzzy intersection.

When a CSP is solved, two solution classes are built: on the one hand, the solutions that satisfy all the constraints and on the other hand, the solutions that violate at least one constraint. With a FCSP, this division of the solution space is refined. First, there are the solutions satisfying completely all constraints ($\mathrm{Sat}(\bar{d}) = 1$); then, the solutions that violate completely at least one constraint ($\mathrm{Sat}(\bar{d}) = 0$); finally, the solutions whose global satisfaction degree is between 0 and 1. The latter set can be ordered according to this degree.

Definition 2. The consistency degree of a FCSP is defined as the satisfaction degree of its best solutions:

$$\begin{aligned}\mathrm{Cons}(\mathscr{P}) &= \sup_{\bar{d} \in \Omega} \mu_{\mathrm{Sols}(\mathscr{P})}(\bar{d}) = \sup_{\bar{d} \in \Omega} \mathrm{Sat}(\bar{d}) \\ &= \sup_{\bar{d} \in \Omega} \min_{C_i \in \mathscr{C}} \mu_{R_i}(\bar{d}).\end{aligned} \tag{11}$$

The so-defined best solutions $\bar{d}_{\mathrm{opt}}$, also called min-optimal, are those whose global satisfaction degree is maximal. Again, they maximize the satisfaction degree of the least satisfied constraints.

The whole thing thus comes down to a max-min-like optimization problem (often called bottleneck).

Let us define the α-cut problem of a fuzzy CSP $\mathscr{P}$ as $\mathscr{P}_\alpha = (\mathscr{X}, \mathscr{D}, \mathscr{C}, \mathscr{R}^\alpha)$ where $\mathscr{R}^\alpha = \{R_1^\alpha, \ldots, R_m^\alpha\}$, and $R_i^\alpha = \{\bar{d}, \mu_{R_i}(\bar{d}) \geqslant \alpha\}$ is the α-level cut of R_i. $\mathscr{P}_\alpha$ is a crisp CSP which is consistent only if $\alpha \leqslant \mathrm{Cons}(\mathscr{P})$.

The most complete arguments for using the min operator when defining the intersection of fuzzy sets have been provided in (Bellman and Giertz, 1973). They adopted a logic point of view, interpreting the intersection of two fuzzy sets as the "logical and" between two statements. They gave reasonable restrictions to be imposed on the intersection operator and proved mathematically that the min operator has to be considered if all properties of Boolean intersections are kept except the excluded middle law and the law of contradiction. Alternative justifications of the min operation exist in other types of problems, for instance in the exploitation of fuzzy preference relations (Pirlot, 1995).

Moreover the min operator is computationally simple, does not involve any assumption on the preference scale and can handle ordinal preferences. Maybe the most relevant properties are that the min operator is idempotent and non-compensatory. In the current framework, we are interested in the maximization of the satisfaction degree of all flexible constraints. No compensation can be accepted between a small degree and a very large one. The preference handling has to express that two middle-range values are better than a small one and a large one. Maximin optimization tends to equalize the satisfaction level of solutions. The preferences are in fact treated in a similar way as vetoes with the maximin approach. As soon as a veto is raised (i.e. a preference is very low), the solution is rejected, whatever is the majority of the other opinions. [3]

2.3. Coping with the drowning effect

Let u_i denote the degree of satisfaction of constraint $C_i \in \mathscr{C}$, i.e., $u_i = \mu_{R_i}(\bar{d})$ and $\bar{u} = (u_1, \ldots, u_m)$. For simplicity in the following, we shall use vectors $\bar{u}$ and $\bar{v}$ instead of solutions $\bar{d}$ and $\bar{d}'$, since when $\bar{d}$ and $\bar{d}'$ are such that $\mu_{R_i}(\bar{d}) = \mu_{R_i}(\bar{d}')\forall i$, they cannot be told apart.

Then, if $\bar{d}$ and $\bar{d}'$ are two min-optimal solutions, associated with vectors $\bar{u}$ and $\bar{v}$, respectively, it holds

$$\min_{i=1,\ldots,m} u_i = \min_{i=1,\ldots,m} v_i \;=\; \mathrm{Cons}(\mathscr{P}).$$

The Drowning Effect is due to the min-based conjunction of the different satisfaction degrees. The "min" operator aggregates the complete information $\mathscr{S}(\bar{u})$ in a unique number: $\mathrm{Sat}(\bar{d})$. The underlying ranking for the solutions is called the *min-ranking*. A solution $\bar{u}$ is better than another $\bar{v}$, if $\mathrm{Sat}(\bar{u})$ is greater than $\mathrm{Sat}(\bar{v})$:

$$\bar{u} >^{\min} \bar{v} \Longleftrightarrow \min_i u_i > \min_i v_i, \tag{12}$$

$$\bar{u} =^{\min} \bar{v} \Longleftrightarrow \min_i u_i = \min_i v_i. \tag{13}$$

This min-ranking is not able to distinguish between possibly numerous optimal solutions. The degrees of satisfaction of constraints other than the most violated ones are not taken into account in the global satisfaction degree. This phenomenon is called the *Drowning Effect*.

Another more basic but weakly committing way of ordering the solutions is the Pareto ordering such that $\bar{u} >^{\text{Pareto}} \bar{v}$ iff $u_i \geqslant v_i, \forall i$, and $u_k > v_k$ for some k. Let us denote by $\mathscr{S}(\bar{d})$ the fuzzy set of constraints satisfied by $\bar{d}$, such that $\mu_{\mathscr{S}(\bar{d})}(C_i) = \mu_{R_i}(\bar{d}) = u_i$. Equivalently, $\bar{u} >^{\text{Pareto}} \bar{v}$ means $\mathscr{S}(\bar{u}) \supset \mathscr{S}(\bar{v})$ for the corresponding solutions $\bar{d}$ and $\bar{d}'$ where $\supset$ denotes fuzzy set inclusion. Any "reasonable" optimal solution to a FCSP should be Pareto-optimal. Yet not all min-optimal solutions are Pareto-optimal, although there is always one that is so (Moulin, 1988; Dubois et al., 1995). The above remarks show that the set of min-optimal solutions needs to be refined in order to get rid of non-Pareto-optimal solutions and overcome the drowning effect.

In the fuzzy literature, no particular attention is given to this problem. In most cases, the min-optimal solution is assumed to be unique and therefore efficient, see Theorem 3.10 or Theorem 4.19 of Sakawa (1993). If in a second step, you are looking for a Pareto-optimal solution, a common trick is to solve a problem with a linear criterion and with lower bounds on satisfaction degrees (Sakawa, 1993; Delgado et al., 1990).

$$\begin{aligned} \max \quad & \sum_{i=1}^{k} \epsilon_i \\ \text{s.t.} \quad & \mu_{R_i}(\bar{x}) - \epsilon_i = \mu_{R_i}(\bar{x}*), \\ & \epsilon_i \geqslant 0, \end{aligned} \tag{14}$$

where $\bar{x}^*$ is your min-optimal solution. In any cases, it gives you an efficient min-optimal solution x. But, this approach is not clearly motivated and seems to be partially inconsistent with the scope of constraint satisfaction since the solution $\bar{x}^*$ is found by a non-compensatory procedure, based on the min-operator. Then, to obtain an efficient solution, Eq. (14) forgets about the non compensatory assumption and uses a linear aggregation. More appropriate procedures have been designed and we will present the relevant ones in the following.

2.3.1. Discrimin-optimality

An intuitively appealing refinement of the conjunction order has been proposed in a logical setting by Brewka (1989) and applied to constraint satisfaction problem by Fargier et al. (1993). It consists in the comparison of two solutions $\bar{u}$ and $\bar{v}$ according to the inclusion of some α-cuts of the fuzzy sets $\mathscr{S}(\bar{u})$ and $\mathscr{S}(\bar{v})$.

[3] This notion of veto can also be used to explain prioritized constraints. The higher is the priority of a constraint, the stronger is its veto and lower is its minimal preference.

Let $\mathscr{S}(\bar{u})_\alpha = \{C_i, u_i \geqslant \alpha\}$. The discrimin ordering is defined as follows:

$$\bar{u} >^{\text{disc}} \bar{v} \Longleftrightarrow \exists \alpha \in [0,1]\colon$$
$$\begin{cases} \forall \beta \in [0,1]\colon \beta < \alpha\colon (\mathscr{S}(\bar{u}))_\beta = (\mathscr{S}(\bar{v}))_\beta, \\ (\mathscr{S}(\bar{u}))_\alpha \supset (\mathscr{S}(\bar{v}))_\alpha, \end{cases} \tag{15}$$

$$\bar{u} =^{\text{disc}} \bar{v} \Longleftrightarrow \forall \alpha \in [0,1]\colon (\mathscr{S}(\bar{u}))_\alpha = (\mathscr{S}(\bar{v}))_\alpha \Longleftrightarrow \mathscr{S}(\bar{u} = \mathscr{S}(\bar{v}). \tag{16}$$

Note that only the α-cuts for small values of α are taken into account. $\bar{u}$ is better than $\bar{v}$ in the sense of discrimin ordering if although the two solutions are feasible for low levels of aspiration, raising this level of aspiration makes $\bar{v}$ infeasible before $\bar{u}$.

$>^{\text{disc}}$ is a partial order. A equivalent definition relies on the set of constraints not respected in the same way by both solutions, $\mathscr{D}(u,v)$ (Dubois et al., 1995):

$$\mathscr{D}(\bar{u},\bar{v}) = \{C_i \in \mathscr{C}\colon v_i \neq u_i\}, \tag{17}$$

$$\bar{u} >^{\text{disc}} \bar{v} \Longleftrightarrow \min_{C_i \in \mathscr{D}(\bar{u},\bar{v})} u_i > \min_{C_i \in \mathscr{D}(\bar{u},\bar{v})} v_i. \tag{18}$$

With the discrimin partial order, we look for *the lowest one among the satisfaction degrees of the constraints not equally satisfied by the two compared solutions*. The discrimin ordering compares the least satisfied discriminating constraints. Eqs. (17) and (18) make it clear that the discrimin ordering refines the min-ordering. The maximal solutions are said to be *discrimin-optimal*. Discrimin-optimal solutions are also Pareto-optimal but not conversely, in general (see Dubois et al., 1995). The discrimin-ordering, like the Pareto one, is only partial, and the relation defined by $\neg(\bar{u} >^{\text{disc}} \bar{v})$ (complement of $>^{\text{disc}}$) is not transitive. If none of $\bar{u} >^{\text{disc}} \bar{v}$ and $\bar{v} >^{\text{disc}} \bar{u}$ hold, it does not follow that $\bar{u} =^{\text{disc}} \bar{v}$.

2.3.2. No reason for regret

Behringer proposed the "No Reason for Regret (NR)" partial ordering in Behringer (1977, 1990) as a refinement of both min- and Pareto-optimalities. He showed that the NR-optimal solutions are min and Pareto optimal, and that the converse is not true in general. As stated in Behringer (1990), $\bar{v}$ is NR-better than $\bar{u}$ if

$$\exists j\colon u_j < v_j \textit{ and } \forall i, (u_j < v_i) \vee (u_i \leqslant v_i).$$

This relation is justified as follows. Assume there is a bundle of "abilities" to be distributed among m members of a government. Let $\bar{u} = (u_1, \ldots, u_m)$ and $\bar{v} = (v_1, \ldots, v_m)$ be two admissible distributions, where u_i is the amount member i will receive in distribution $\bar{u}$. Every i tries to be as satisfied ($u_i = \mu_{R_i}(\bar{u})$) as possible. There is a collective mechanism (unanimity, veto …) which generates a feeling for equity or fairness. Distribution $\bar{v}$ will be preferred if:

1. At least one member feels better with $\bar{v}$

   $\exists j\colon u_j < v_j$

   and
2. all remaining members feel at least as good

   $\forall i\colon\ u_i \leqslant v_i$

   or
3. for those i's who receive less satisfaction ($v_i < u_i$), i has "no reason to regret" the choice of distribution $\bar{v}$ instead of $\bar{u}$, because member i is still better than member j was with distribution $\bar{u}$

   $u_j < v_i$.

   With distribution $\bar{u}$, j would have grounds on which to complain and might put his veto, disabling the government agreement.

We show now that the NR-order relation proposed by Behringer (1977) is equivalent to the discrimin relation.

Theorem 1.

$$\bar{v} >^{\text{disc}} \bar{u} \Longleftrightarrow \bar{v} >^{\text{NR}} \bar{u}.$$

Proof. $\underline{>^{\text{disc}} \Rightarrow >^{\text{NR}}}$

$$\bar{v} >^{\text{disc}} \bar{u} \Longleftrightarrow \min_{C_i \in \mathscr{D}(\bar{u},\bar{v})} v_i > \min_{C_i \in \mathscr{D}(\bar{u},\bar{v})} u_i,$$

where $\mathscr{D}(\bar{u},\bar{v}) = \{C_i \in \mathscr{C}\colon u_i \neq v_i\}$.

Let u_l (resp. v_k) be the minimum of u_i (resp. v_i) on $\mathscr{D}(\bar{u},\bar{v})$. Therefore, we have

$$\exists\, C_l, \quad u_l < v_k \leqslant v_l,$$

$$\forall C_i \notin \mathscr{D}(\bar{u},\bar{v}), \quad u_i \leqslant v_i,$$

$$\forall C_i \in \mathscr{D}(\bar{u},\bar{v}), \quad u_l < v_k \leqslant v_i.$$

And $\bar{v} >^{\mathrm{NR}} \bar{u}$.

$\underline{>^{\mathrm{NR}} \Rightarrow >^{\mathrm{disc}}}$

$$\bar{v} >^{\mathrm{NR}} \bar{u} \Longleftrightarrow \exists j{:}\, u_j < v_j, \text{ and}$$

$$\forall i{:}\, (u_j < v_i) \vee (u_i \leqslant v_i).$$

Therefore, for all C_i in $\mathscr{D}(\bar{u},\bar{v})$, we have either $u_j < v_i < u_i$ or $u_i < v_i$. Let $\mathscr{D}_1 = \{C_i \in \mathscr{D}(\bar{u},\bar{v}), u_j < v_i < u_i\}$ and $\mathscr{D}_2 = \{C_i \in \mathscr{D}(\bar{u},\bar{v}), u_i < v_i\}$. $(\mathscr{D}_1, \mathscr{D}_2)$ is a partition of $\mathscr{D}(\bar{u},\bar{v})$. $\mathscr{D}(\bar{u},\bar{v})$ is not empty since $C_j \in \mathscr{D}_2$. Then

$$\min_{C_i \in \mathscr{D}_1} u_i > \min_{C_i \in \mathscr{D}_1} v_i > u_j$$

and

$$\min_{C_i \in \mathscr{D}_2} u_i \leqslant u_j \text{ since } C_j \in \mathscr{D}_2$$

hence

$$\min_{C_i \in \mathscr{D}(\bar{u},\bar{v})} u_i = \min_{C_i \in \mathscr{D}_2} u_i \leqslant u_j$$

Now

$$\min_{C_i \in \mathscr{D}_1} v_i > u_j \geqslant \min_{C_i \in \mathscr{D}(\bar{u},\bar{v})} u_i$$

and

$$\min_{C_i \in \mathscr{D}_2} v_i > \min_{C_i \in \mathscr{D}_2} u_i \geqslant \min_{C_i \in \mathscr{D}(\bar{u},\bar{v})} u_i$$

hence

$$\min_{C_i \in \mathscr{D}(\bar{u},\bar{v})} v_i > \min_{C_i \in \mathscr{D}(\bar{u},\bar{v})} u_i$$

And $\bar{v} >^{\mathrm{disc}} \bar{u}$. □

The two partial orders are thus mathematically equivalent. But the interpretation is easier with Discrimin, based either on the inclusion of α-cuts or on the difference set $\mathscr{D}(\bar{u},\bar{v})$.

It should be stated that *discrimin-maximal* makes more sense than *discrimin-optimal*, since the discrimin relation is not a total order. Nevertheless for the sake of simplicity and uniformity, we will use the latter.

2.3.3. Leximin ranking-Leximin optimality

The previous refinement of the min-ordering is an approach based on the inclusion of α-cuts. The leximin order relies on the cardinality of these cuts. A solution $\bar{u}$ will be preferred to another $\bar{v}$, according to the leximin order, if, there is a threshold α such that for all $\beta < \alpha$, the number of constraints satisfied by $\bar{u}$ at level at least β is equal to the number of constraints satisfied by $\bar{v}$, but $\bar{u}$ satisfies more constraints than $\bar{v}$ at level α. In other words, this is definition Eqs. (15) and (16), where cardinalities of level cuts are compared. It checks the cardinality of the α-cuts for the fuzzy sets $\mathscr{S}(\bar{u})$ and $\mathscr{S}(\bar{v})$, by increasing value of α. As soon as a difference occurs, the solution whose cut has the highest cardinality is chosen.

In this framework, we keep the complete information conveyed by $\mathscr{S}(\bar{u})$ but we rank the different satisfaction degrees in an increasing order. Let the components of vector $\bar{u}$ be ranked such that $u_{i_1} \leqslant u_{i_2} \leqslant \cdots \leqslant u_{i_m}$ and let $\bar{u}^*$ be the vector such that $u^*_j = u_{i_j}$. The leximin ordering $>^{\mathrm{lm}}$ between vectors $\bar{u}$ and $\bar{v}$ is defined on the basis of the vectors $\bar{u}^*$ and $\bar{v}^*$ with reordered components, ranked by means of the standard lexicographic order $>^{\mathrm{lex}}$ (Fishburn, 1974):

$$\bar{u} >^{\mathrm{lm}} \bar{v} \Longleftrightarrow \bar{u}^* >^{\mathrm{glex}} \bar{v}^* \tag{19}$$

$$\Longleftrightarrow \exists i \leqslant k{:} \begin{cases} \forall j < i{:}\, u^*_j = v^*_j, \\ u^*_i > v^*_i; \end{cases} \tag{20}$$

$$\bar{u} =^{\mathrm{lm}} \bar{v} \Longleftrightarrow \forall i \leqslant k{:}\, u^*_i = v^*_i \tag{21}$$

$$\Longleftrightarrow \neg(\bar{u} >^{\mathrm{lm}} \bar{v}) \wedge \neg(\bar{v} >^{\mathrm{lm}} \bar{u}). \tag{22}$$

$>^{\mathrm{lm}}$ is also a ranking, since $=^{\mathrm{lm}}$ is transitive.

It can be shown (Fargier, 1994) that this ranking leads to the solutions violating the smallest number of fuzzy constraints, in the sense of a fuzzy-valued cardinality. The different orderings have been presented in increasing order of refinement: solutions being incomparable for one order may be distinguished by the next one:

$$\bar{u} >^{\mathrm{lm}} \bar{v} \Rightarrow \bar{u} \geqslant^{\mathrm{disc}} \bar{v} \Rightarrow \bar{u} \geqslant^{\mathrm{min}} \bar{v}. \tag{23}$$

Also, any leximin-optimal solution is discrimin-optimal.

Our presentation of the Leximin-optimality comes from the Artificial Intelligence literature (Benferhat et al., 1993; Fargier et al., 1993; Dubois et al., 1996). In the context of social choice it can be found in Moulin (1988) and Sen (1986). In Operation Research it has been studied by Behringer (1981) It relies on min- and Pareto-optimality. An older approach to leximin, known in Numerical Analysis as "strict Chebyshev norm", has been introduced by Rice (1962) and Descloux (1963). In the spirit of this last approach, some additional results have been proved in Dubois et al. (1997).

Classical to the fuzzy field are t-norms and t-conorms. A t-norm T is a semigroup of the unit interval (associative, commutative, with identity 1) which is non-decreasing in each place. Archimedean t-norms are such that $\forall a \in (0,1)$, $aTa < a$. Any Archimedean continuous t-norm T can be written as follows (Schweizer and Sklar, 1983):

$$aTb = \phi^{-1}(\min(\phi(0), \phi(a) + \phi(b))),$$

where $\phi\colon [0,1] \to [0, \phi(0)]$ is a continuous and decreasing function, such that $\phi(1) = 0$. Similar to Hölder norms, families of t-norms T_p can be written as

$$T_p(\bar{x}) = \phi_p^{-1}\left(\sum_{i=1}^{n} \phi_p(x_i)\right) \tag{24}$$

for which $\phi_p(0) = +\infty$. For our purpose, we only consider parameterized increasing families of t-norms such that

$$\lim_{p\to\infty} T_p(\bar{x}) = |\bar{x}|_\infty = \min_i x_i,$$

e.g., the Frank family ($q = 1/p$):

$$T_p(\bar{x}) = \log_q\left[1 + \frac{\Pi_{i=1}^{n}(q^{x_i} - 1)}{(q-1)^{(n-1)}}\right]. \tag{25}$$

Among other things, it is proved in Dubois et al. (1997) that in a convex set, the solutions of the maximization with respect to Archimedean triangular norm-based ordering converge to the solution of the maximization of the leximin-based ordering, as soon as the triangular norm converges to the minimum operator. The same results hold for the family of generalized arithmetic means of the form

$$f_p(x_1, \ldots, x_i) = \phi_p^{-1}\left(\frac{1}{i}\sum_{j=1}^{i} \phi_p(x_j)\right), \tag{26}$$

where ϕ_p is any continuous strictly monotonic function on $[0,1]$; and for the ordered weighted average operations proposed by Yager (1988) as well.

These results can be considered as another argument in favour of the leximin procedure, as the leximin appears as the limit of parametered families of very common aggregation operators, namely t-norms, generalized means and OWAs.

In the next section, we illustrate the notions of min, discrimin and leximin on a project scheduling problem. Some observations about the critical path will give hints about a constructive procedure for getting the optimal solution according the different orderings.

3. Scheduling problem with flexible constraints

Consider the project scheduling problem (Hillier and Lieberman, 1989). This problem is represented by a graph, like in Fig. 1. The edges encode the precedence constraints between the operations depicted by the nodes. The value of each node is the duration of the corresponding operation. The scheduling problem requires that all the starting times be computed.

This problem with crisp data (durations, ready or due dates) has been extensively studied, as well

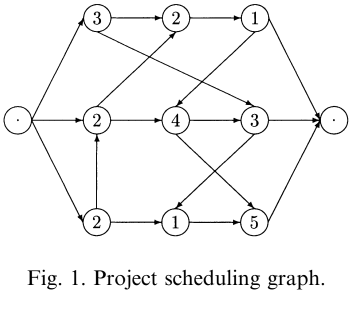

Fig. 1. Project scheduling graph.

as its stochastic counterpart (e.g. PERT-CPM methods (Hillier and Lieberman, 1989)). But, the operation durations may be either not-precisely known or flexible. In the first case durations are not completely known and not controllable. In the second case, durations are not known because they have not been decided yet. For example, if the speed of an operation is tunable, it may be better to allow for a longer duration leading to an increased quality of the result (cf. Fig. 2). Such preferences may also exist about the ready date (Fig. 3) and the due date (Fig. 4). The customer prefers to be supplied as soon as possible, but not later than a given deadline. These constraints about duration, ready date and due date are *flexible*. There are a matter of preference, not of randomness.

3.1. Statement of the problem

Let us assume that the temporal parameters are flexible and assume independent fuzzy specifications for the ready date ($\widetilde{\mathrm{RD}}_i$), the duration ($\widetilde{\mathrm{DU}}_i$) and the due date ($\widetilde{\mathrm{DD}}_i$). The set of decision variables is more complex than in the classical case. It includes the starting date (s_i) and the duration (u_i) of every operation (Dubois et al., 1995; Fargier, 1994).

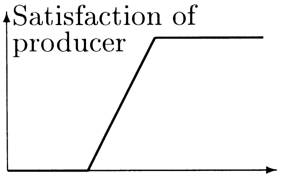

Fig. 2. Ready date.

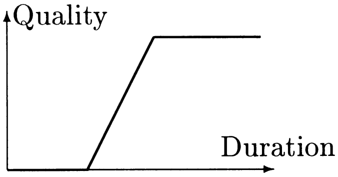

Fig. 3. Flexible duration.

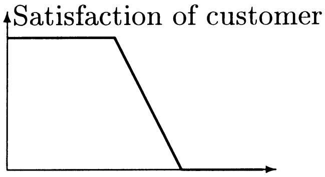

Fig. 4. Due date.

Some constraints remain crisp. The precedence constraints (as well as the capacity constraints, if any) are still represented by a crisp relation in $\mathscr{R}$. As usual, we have $\forall k \in \Gamma_i$ (where Γ_i is the set of indices of the operations preceded by O_i)

$$s_k \geqslant s_i + u_i. \tag{27}$$

But the other constraints are flexible: for instance the duration of operation O_i should be *near* 3 h, O_i should last *at least* 1 h and *not more than* 3 h. All those words in *italics* represent flexible parameters.

The initial temporal window where the operation O_i has to be performed is the fuzzy interval $[\widetilde{\mathrm{RD}}_i, \widetilde{\mathrm{DD}}_i]$ pictured on Fig. 5.

The flexible constraints of this problem can be easily obtained from the relations (8) and (9).

$$s_i \in [\widetilde{\mathrm{RD}}_i, \widetilde{\mathrm{DD}}_i \ominus \widetilde{\mathrm{DU}}_i], \tag{28}$$

$$u_i \in \widetilde{\mathrm{DU}}_i, \tag{29}$$

$$s_k \geqslant s_i + u_i \quad \forall k \in \Gamma_i \tag{30}$$

where $\ominus$ denotes the fuzzy subtraction of two fuzzy numbers (see Dubois and Prade, 1987).

The global satisfaction degree $\mathrm{Sat}(\bar{d})$ is

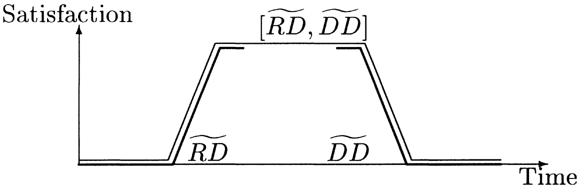

Fig. 5. Fuzzy temporal window.

$$\mathrm{Sat}(\bar{d})$$

$$= 0 \quad \text{(if a precedence constraint (30) is violated)} \tag{31}$$

$$= \min_{O_i}\left(\min\left\{ \mu_{[\widetilde{\mathrm{RD}}_i, \widetilde{\mathrm{DD}}_i \ominus \widetilde{\mathrm{DU}}_i]}(s_i), \mu_{\widetilde{\mathrm{DU}}_i}(u_i) \right\}\right). \tag{32}$$

In general, the fuzzy durations are represented by trapezoidal fuzzy numbers, see Fig. 6.

Some simplifications can be achieved according to the meaning of the flexibility. In this paper, we will only study the case of tunable operations. For the unknown durations case and for some links between the two interpretations, we refer the reader to (Fargier, 1994; Dubois et al., 1995).

For each value of the satisfaction degree, only the shortest duration is meaningful. As a matter of fact, if two durations have the same degree of satisfaction, the shortest one will give more time to other operations in the project to be performed. Therefore, only the left part of the fuzzy number $\widetilde{\mathrm{DU}}_i$ will be considered:

$$u_i \in [\widetilde{\mathrm{DU}}_i, +\infty). \tag{33}$$

The satisfaction of constraints (29) and (30) are simplified in this context (see Dubois et al., 1995). Indeed, the duration must be set such that $u_i \leqslant \min_{k \in \Gamma_i}(s_k - s_i)$. Since the membership function of $[\widetilde{\mathrm{DU}}_i, +\infty)$ is increasing, we choose $u_i = \min_{k \in \Gamma_i}(s_k - s_i)$ and the satisfaction degree of the constraint is

$$\min_{k \in \Gamma_i} \mu_{[\widetilde{\mathrm{DU}}_i, +\infty)}(s_k - s_i).$$

The global satisfaction degree is

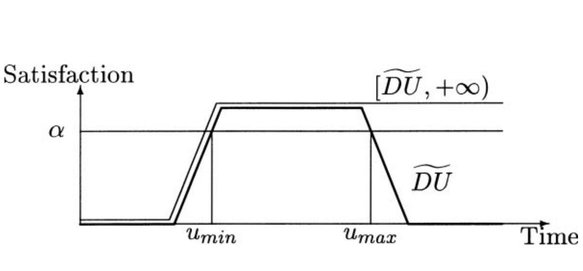

Fig. 6. Flexible duration and $[\widetilde{\mathrm{DU}}_i, +\inf]$.

$$\mathrm{Sat}(\bar{d}) = 0$$

$$\text{(if a precedence constraint (30) is violated)}$$

$$= \min_{O_i}\left(\min\left\{ \mu_{[\widetilde{\mathrm{RD}}_i, \widetilde{\mathrm{DD}}_i \ominus \widetilde{\mathrm{DU}}_i]}(s_i), \min_{k \in \Gamma_i} \mu_{[\widetilde{\mathrm{DU}}_i, +\infty)}(s_k - s_i) \right\}\right). \tag{34}$$

Note that the variables u_i seem to disappear. In fact, they are implicitly computed and will be chosen according to the starting times obtained from the problem (34). The latter is equivalent to problem (32), regarding the computation of the degree of consistency of the fuzzy CSP.

The fuzzy PERT problem has been considered by many authors for a long time (Dubois and Prade, 1978; Chanas and Kamburowski, 1981; Dubois and Prade, 1987; Lootsma, 1989; Fortemps, 1997). However in these works, the meaning of the fuzzy numbers is often different from here, in the sense that fuzzy durations model uncertainty (Slowiński and Teghem, 1990) rather than preference. Here only preference profiles are considered and the result is a precise solution obtained from "optimal defuzzification".

3.2. Propagation of flexible constraints – Consistency degree

As for the classical problem, we also use constraint propagation so as to reduce the temporal window of each operation.

The forward propagation step says that an operation may begin neither before its "ready date" nor before the latest ending time of the preceding operations. For each operation, this latest ending time is computed as the sum of the starting time and the duration of the operation. The backward propagation step prescribes that an operation finishes before its "due date", as well as before the earliest starting time of the following operations.

Therefore, for every pair (i, j) such that i precedes j $(j \in \Gamma_i)$, we update ready dates and latest ending dates of operations as follows:

$$\widetilde{rd}_j := \widetilde{\max}(\widetilde{rd}_j, \widetilde{rd}_i \oplus \widetilde{\mathrm{DU}}_i), \tag{35}$$

$$\widetilde{dd}_i := \widetilde{\min}(\widetilde{dd}_i, \widetilde{dd}_j \ominus \widetilde{\mathrm{DU}}_j), \tag{36}$$

and the propagation of both kinds of constraints gives the fuzzy temporal window allowed for the performing of the operation. Initially, $\widetilde{rd}_j = \widetilde{\mathrm{RD}}_j$ and $\widetilde{dd}_j = \widetilde{\mathrm{DD}}_j$. Finally, by subtracting the fuzzy duration $\widetilde{\mathrm{DU}}_i$ from the latest ending time $\widetilde{dd}_i$, we obtain the (fuzzy) latest starting time. The values which best satisfy both starting times (soonest and latest ones) give the temporal slack for the starting time of O_i, namely

$$\widetilde{sl}_i = [\widetilde{rd}_i, \widetilde{dd}_i \ominus \widetilde{\mathrm{DU}}_i]. \tag{37}$$

Note that the fuzzy subtraction increases the imprecision in the sense that $\widetilde{dd}_i \ominus \widetilde{\mathrm{DU}}_i$ is more imprecise than $\widetilde{dd}_i$ or $\widetilde{\mathrm{DU}}_i$. It makes sense because the computation of $\widetilde{dd}_i$ involves only operations after O_i and the independent fuzzy overall due-date. Hence $\widetilde{dd}_i$ and $\widetilde{\mathrm{DU}}_i$ are non-interactive (see Fig. 7).

It is easy to verify that

$$\begin{aligned}\mathrm{Cons}(\mathscr{P}) &= \sup_{s_1,\ldots,s_n} \min_{i=1,\ldots,n} \mu_{\widetilde{sl}_i}(s_i) \\ &= \min_{i=1,\ldots,n} \mathrm{height}(\widetilde{sl}_i).\end{aligned}$$

Indeed, if $\mathrm{height}(\widetilde{sl}_i) = \alpha_i < 1$, then the choice of s_i^* such that

$$\mu_{\widetilde{rd}_i}(s_i^*) = \mu_{\widetilde{dd}_i \ominus \widetilde{\mathrm{DU}}_i}(s_i^*)$$

is unique and it leads to a best choice of the duration u_i^* such that

$$\alpha_i = \mu_{\widetilde{[\mathrm{DU}_i, +\infty)}}(u_i^*).$$

Let $\{O_1, \ldots, O_k\}$ be the set of operations such that $\mathrm{height}(\widetilde{sl}_i) = \mathrm{Cons}(\mathscr{P})$. These operations correspond to a *set of critical paths*, and can be called *critical operations*. For any two consecutive operations along a critical path, it is easy to check that $s_i^* - s_j^* = u_j^*$ so that the term $\min_{k \in \Gamma_i} \mu_{\widetilde{[\mathrm{DU}_i,+\infty)}}(s_k - s_i)$ in (34) is redundant. By definition some slack exists for the choice of starting times of non-critical operations. But due to the preference profiles computed from the fuzzy numbers, the choice of these starting times will affect the discrimin-optimality of the selected solution. In particular, choosing $s_i = s_i^*$ induced by $\mathrm{height}(\widetilde{sl}_i)$ is not the best choice for non-critical operations.

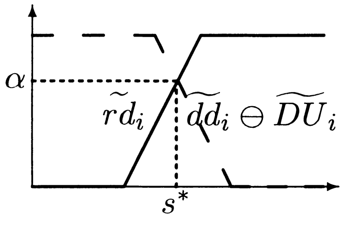

Fig. 7. The slack and the starting time.

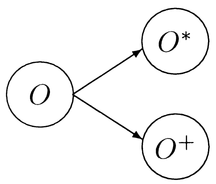

Fig. 8. Problem graph.

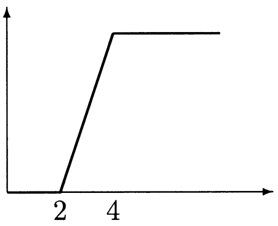

Fig. 9. Common duration.

Example 1. Let 3 operations O, O^* and O^+ (Fig. 8) where operation O precedes the two others. Their flexible duration is the same $\widetilde{\mathrm{DU}}$: $(2, 4, \infty, \infty)$ [4] depicted on Fig. 9. The ready date $\widetilde{\mathrm{RD}}$ of each operation is zero. The due dates are $\widetilde{\mathrm{DD}} = (-\infty, -\infty, 8, 8)$, $\widetilde{\mathrm{DD}}^* = (-\infty, -\infty, 4, 8)$ and $\widetilde{\mathrm{DD}}^+ = (-\infty, -\infty, 6, 8)$ (see Fig. 10). The constraints are:

[4] We use here trapezoidal fuzzy numbers. Thus, we have the lower bounds respectively of the cut at $\alpha = 0$ and $\alpha = 1$ followed by the upper bounds respectively for the cut at $\alpha = 1$ and $\alpha = 0$.

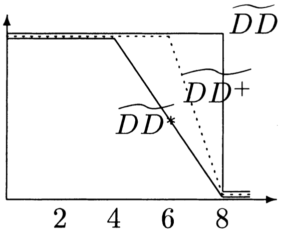

Fig. 10. “Due dates”.

$$s \in [\widetilde{\mathrm{RD}}, \widetilde{\mathrm{DD}} \ominus \widetilde{\mathrm{DU}}], \quad u \in \widetilde{\mathrm{DU}},$$
$$s^* \in [\widetilde{\mathrm{RD}}^*, \widetilde{\mathrm{DD}}^* \ominus \widetilde{\mathrm{DU}}^*], \quad u^* \in \widetilde{\mathrm{DU}}^*,$$
$$s^+ \in [\widetilde{\mathrm{RD}^+}, \widetilde{\mathrm{DD}^+} \ominus \widetilde{\mathrm{DU}^+}], \quad u^+ \in \widetilde{\mathrm{DU}^+},$$
$$s^* \geqslant s + u, \quad s^+ \geqslant s + u.$$

The global satisfaction degree of a solution is written:

$$\begin{aligned}
\mathrm{Sat}(\bar{d}) &= \mathrm{Sat}(s, s^*, s^+, u, u^*, u^+) \\
&= \min\Big(\min\left\{ \mu_{[0,+\infty)}(s), \mu_{[0,+\infty)}(s^*), \mu_{[0,+\infty)}(s^+) \right\}, \\
&\quad \min\left\{ \mu_{\widetilde{\mathrm{DU}}}(u), \mu_{\widetilde{\mathrm{DU}}}(u^*), \mu_{\widetilde{\mathrm{DU}}}(u^+) \right\}, \\
&\quad \min\Big\{ \mu_{(-\infty,\widetilde{\mathrm{DD}}]}(s + u), \\
&\qquad \mu_{(-\infty,\widetilde{\mathrm{DD}^*}]}(s^* + u^*), \mu_{(-\infty,\widetilde{\mathrm{DD}^+}]}(s^+ + u^+) \Big\} \Big).
\end{aligned}$$

We follow the propagation steps (35) and (36). Using the slack time defined in Eq. (37), we have:

$$\widetilde{rd} = [0, 0, \infty, \infty), \quad \widetilde{dd} \ominus \widetilde{\mathrm{DU}} = (-\infty, \infty, -4, +4],$$
$$\widetilde{rd}^+ = [2, 4, \infty, \infty), \quad \widetilde{dd}^+ \ominus \widetilde{\mathrm{DU}}^+ = (-\infty, -\infty, 2, 6],$$
$$\widetilde{rd}^* = [2, 4, \infty, \infty), \quad \widetilde{dd}^* \ominus \widetilde{\mathrm{DU}}^* = (-\infty, -\infty, 0, 6],$$

and

$$\mathrm{height}(\widetilde{sl}) = \frac{1}{2}, \quad \mathrm{height}(\widetilde{sl}^+) = \frac{2}{3},$$
$$\mathrm{height}(\widetilde{sl}^*) = \frac{1}{2}.$$

By constraint propagation, the different fuzzy numbers are reduced. When a stable state is reached, the problem consistency degree is the height of the lowest fuzzy number. In our example, this degree is equal to $\frac{1}{2}$.

The basic solution should be to take all the durations equal to 3 (as induced by the degree $\frac{1}{2}$). And the corresponding starting times are easily computed: $s = 0$, $s^* = 3$ and $s^+ = 3$. The satisfaction degrees vector is

$$\begin{aligned}
&\Big(\mu_{\widetilde{\mathrm{DU}}}(u), \mu_{\widetilde{\mathrm{DU}}}(u^*), \mu_{\widetilde{\mathrm{DU}}}(u^+), \mu_{\widetilde{\mathrm{DD}}}(s + u), \\
&\quad \mu_{\widetilde{\mathrm{DD}}^*}(s^* + u^*), \mu_{\widetilde{\mathrm{DD}}^+}(s^+ + u^+) \Big) \\
&= \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1, \frac{1}{2}, 1 \right). \qquad (38)
\end{aligned}$$

The following choice of starting times and durations (based on the core of the slacks) $s = 0$, $s^* = 3$ and $s^+ = 3.333$; and $u = 3$, $u^* = 3$ and $u^+ = 3.333$ leads to a satisfaction degrees vector:

$$\begin{aligned}
&\Big(\mu_{\widetilde{\mathrm{DU}}}(u), \mu_{\widetilde{\mathrm{DU}}}(u^*), \mu_{\widetilde{\mathrm{DU}}}(u^+), \mu_{\widetilde{\mathrm{DD}}}(s + u), \\
&\quad \mu_{\widetilde{\mathrm{DD}}^*}(s^* + u^*), \mu_{\widetilde{\mathrm{DD}}^+}(s^+ + u^+) \Big) \\
&= \left(\frac{1}{2}, \frac{1}{2}, \frac{2}{3}, 1, \frac{1}{2}, \frac{2}{3} \right). \qquad (39)
\end{aligned}$$

This second solution is considered as a better compromise, as it satisfies three constraints with a degree greater than $\frac{1}{2}$. The consistency degree of this solution remains $\frac{1}{2}$.

This small problem suffers from the Drowning Effect. The two solutions found have the same satisfaction degree and cannot be distinguished by the min-ordering, even if one of them appears better.

3.3. Improving the solution

Among the classical algorithms for scheduling problems, the Critical Path Method is crucial (Hillier and Lieberman, 1989). The critical path is, through the problem graph, a path from a ready date node to a due date one such that, for each node of this path, the starting time of the related operation is equal to the ending time of the previous node operation, Along the critical path, all the constraints are saturated; there is no slack.

In a scheduling problem with flexible durations, it may appear also one or several critical paths, in the α-cut problem corresponding to $\alpha = \mathrm{Cons}(\mathscr{P})$.

Therefore, in the previous example, considering the $\frac{1}{2}$-cut, the path O–O^* is critical. Along this critical path, the variables values are enforced by the corresponding α-cut of the fuzzy numbers. There is no slack. There is no other choice of better (and longer) durations.

The algorithm we propose to improve the already found solutions is recursive. From the complete problem graph, we compute the problem consistency degree (by constraint propagation). As a by-product, we obtain the critical paths. Along these paths, we can instantiate the variables (according to the cut $\alpha = \mathrm{Cons}(\mathscr{P})$. Then, with a reduced graph – since several variables are now determined and no longer "variable" –, we can again apply the same principle: look for the critical path and instantiate.

If we apply this algorithm to our example, we obtain the following.

Example 1 (continued). The problem consistency degree equals to $\frac{1}{2}$. Therefore, we will study the cut $\alpha = \frac{1}{2}$ of the problem.

$$(s = 0) + (u = 3) < 8,$$
$$(s = 0) + (u = 3) + (u^* = 3) = 6,$$
$$(s = 0) + (u = 3) + (u^+ = 3) < 7.$$

The path O–O^* is critical. We can deduce the value of s and u, s^* and u^* ($s = 0$, $u = 3$, $s^* = s + u = 3$, $u^* = 3$).

By the way, s^+ is automatically obtained ($s^+ = s + u = 3$). The simplified problem where only u^+ remains to be determined has the consistency degree $\frac{3}{4} = \mu_{\widetilde{\mathrm{DD}}^+ \ominus \widetilde{\mathrm{DU}}}(3)$. We can therefore use the value $u^+ = 3.5$ (see Fig. 11).

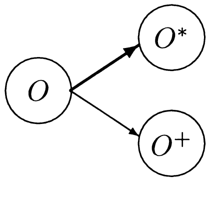

Fig. 11. Operations graph and critical path $O - O^*$ (in bold).

The satisfaction degrees vector is:

$$\Big(\mu_{\widetilde{\mathrm{DU}}}(u), \mu_{\widetilde{\mathrm{DU}}}(u^*), \mu_{\widetilde{\mathrm{DU}}}(u^+), \mu_{\widetilde{\mathrm{DD}}}(s + u),$$
$$\mu_{\widetilde{\mathrm{DD}}^*}(s^* + u^*), \mu_{\widetilde{\mathrm{DD}}^+}(s^+ + u^+)\Big)$$
$$= \left(\frac{1}{2}, \frac{1}{2}, \frac{3}{4}, 1, \frac{1}{2}, \frac{3}{4}\right). \tag{40}$$

This multi-step constraint propagation gives a solution really better than the two others. It is the unique discrimin-optimal solution to the problem, as proved in the sequel.

4. General algorithms for computing discrimin and leximin solutions

In this section, we extend the intuitive approach of the previous section to a more general framework. We outline a multi-step constraint satisfaction procedure which yields all discrimin-optimal and leximin-optimal solutions. This result is consistent with the original algorithmic definition of Leximin by Rice (1962).

4.1. Critical subsets of constraints

The main observation from the previous section is the existence of critical paths for a given α-cut of the fuzzy scheduling problem, namely with $\alpha = \mathrm{Cons}(\mathscr{P})$. By solving the saturated constraints, we have been able to define a new subproblem whose solution is better than the one obtained by a defuzzification at level $\mathrm{Cons}(\mathscr{P})$.

We first need to recall the definition of the strong α-cut and the α-section of a fuzzy subset $\tilde{R}$:

Definition 3. The strong α-cut of a fuzzy subset $\tilde{R}$ is the crisp subset

$$R_\alpha^> = \left\{\bar{d}\colon \mu_{\tilde{R}}(\bar{d}) > \alpha\right\}.$$

Definition 4. The α-section of a fuzzy subset $\tilde{R}$ is the crisp subset

$$R_\alpha^= = \left\{\bar{d}\colon \mu_{\tilde{R}}(\bar{d}) = \alpha\right\}.$$

We give now a wider definition of criticity:

Definition 5. $\mathcal{C}'$ is a subset of saturated constraints of the FCSP $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C}, \mathcal{R})$ if and only if there exists a solution $\bar{d} \in \Omega$ such that

$$\forall C_i \in \mathcal{C}', \quad \mu_{R_i}(\bar{d}) = \alpha^*,$$

$$\forall C_i \notin \mathcal{C}', \quad \mu_{R_i}(\bar{d}) > \alpha^*,$$

and $\alpha^* = \text{Cons}(\mathcal{P})$.

In the sequel, we shall work with *minimal subsets of saturated constraints*, according to two different notions of minimality. We denote by $2^{\mathcal{C}}$ the set of subsets of $\mathcal{C}$ and $\mathcal{C}\mathcal{C}$ the set of saturated subsets of $\mathcal{C}$. Obviously, $\mathcal{C}\mathcal{C} \subseteq 2^{\mathcal{C}}$.

Definition 6. A subset $A \in \mathcal{C}\mathcal{C}$ is critical if and only if it is inclusion-minimal in $\mathcal{C}\mathcal{C}$, that is it does not exist a subset $B \in \mathcal{C}\mathcal{C}$ such that $B \subset A$. A constraint in a critical subset is said to be critical.

Definition 7. A subset $A \in \mathcal{C}\mathcal{C}$ is cardinality-minimal if and only if it does not exist a subset $B \in \mathcal{C}\mathcal{C}$ such that $|B| < |A|$.

Clearly a minimal cardinality subset of $\mathcal{C}\mathcal{C}$ is critical. If a subset A of constraints is critical then consider the problem $\mathcal{P}' = (\mathcal{X}, \mathcal{D}, \mathcal{C}, \mathcal{R}')$ where $\mathcal{R}' = \{R_i \cap R^{>}_{i_\alpha}, C_i \notin A\} \cup \{R^{=}_{i_\alpha}, C_i \in A\}$ with $\alpha = \text{Cons}(\mathcal{P})$. Note that the set $\bigcap_{C_i \in \mathcal{C}} R^{>}_{i_\alpha}$ is empty, i.e. the corresponding classical CSP has no solutions, by definition of $\text{Cons}(\mathcal{P})$.

For any critical subset of constraints A, $\mathcal{C} \setminus A$ is a maximal consistent subset of constraints of the form $\bar{d} \in R^{>}_{i_\alpha}$ in $\mathcal{C}$. So, problem $\mathcal{P}'$ always have solutions.

An important special case of fuzzy CSP is when fuzzy constraints are unary, i.e. only variables have fuzzy domains, while other constraints, involving several variables are all crisp. In that case, a critical subset of constraints is actually a critical subset of decision variables. If $F_1, \ldots, F_n$ are the fuzzy domains of $X_1, \ldots, X_n$, and A is a critical subset of variables then the set of values $F^{=}_{i_\alpha}$ of $X_i \in A$ is generally very small, so that the problem $\mathcal{P}'$ resulting from selecting a critical subset of variables is generally much simpler than $\mathcal{P}$.

4.2. A discrimin-optimal algorithm

Algorithm DA

Find the discrimin solutions of $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C}, \mathcal{R})$

1. Compute the optimal satisfaction degree, i.e. the consistency degree

   $$\alpha^* = \text{Cons}(\mathcal{P}) = \sup_{\bar{d} \in \Omega} \min_{C_i \in \mathcal{C}} \mu_{R_i}(\bar{d}).$$

2. Determine the critical subsets of constraints $\mathcal{C}^*_k$ $k = 1, \ldots$ (i.e. minimal according to the set-inclusion).
3. For every critical set $\mathcal{C}^*_k$.

   Find the discrimin solutions of $\mathcal{P}' = (\mathcal{X}, \mathcal{D}, \mathcal{C}'_k, \mathcal{R}'_k)$ where

- each constraint $C_i \in \mathcal{C}^*_k$ has been transformed in the classical constraint $C'_i \in \mathcal{C}'_k$ associated with the α^*-section

  $$R'_i = R^{=}_{i_\alpha},$$

- each constraint $C_i \notin \mathcal{C}^*_k$ is kept with the restriction that its satisfaction degree has to be greater than α^*:

  $$R'_i = R_i \cap R^{>}_{i_\alpha}.$$

The algorithm terminates in a finite number of steps because some constraints are defuzzified (replaced by their α-cut) at each step. We denote $\{\mathcal{P}^{(l)}\}$, $l = 0, \ldots$, a sequence of reductions of the initial problem. As a corollary of the algorithm definition, we have

$$\text{Cons}(\mathcal{P}^{(l)}) > \text{Cons}(\mathcal{P}^{(l-1)}).$$

The strict inequality holds when a complete "critical subset" is used. In some cases, it can be easier to defuzzify only one critical constraint at a time and not the whole critical subset. Then, we still have the following property, which is sufficient for the proof of optimality:

$$\text{Cons}(\mathcal{P}^{(l)}) \geqslant \text{Cons}(\mathcal{P}^{(l-1)}).$$

Theorem 2. *A solution $\bar{d}$ is discrimin-optimal if and only if $\bar{d}$ is given by the algorithm DA.*

Proof. If $\bar{d}$ is discrimin optimal, then $\forall C_j \in \mathcal{C}$ the set of crisp constraints:

$$\mu_{R_i}(\bar{d}') \geqslant \mu_{R_i}(\bar{d}),$$

$$\mu_{R_j}(\bar{d}') > \mu_{R_j}(\bar{d}),$$

is inconsistent. If $\alpha_1 < \alpha_2 < \cdots < \alpha_p$ are the set of distinct satisfaction levels in $\{\mu_{R_i}(\bar{d}), C_i \in \mathscr{C}\}$, and $\mathscr{C}_k = \{C_i \in \mathscr{C}: \mu_{R_i}(\bar{d}) = \alpha_k\}$. It is obvious that for any $k = 1, \ldots, p$, there exists $\mathscr{P}^{(k)}$ such that $\bar{d}$ is a solution of $\mathscr{P}^{(k)}$ and such that $\mathscr{C}_k$ is a set of critical constraints for the same problem. Hence $\bar{d}$ is found by the algorithm.

Conversely, if $\bar{d}$ is not discrimin optimal, then there exists $\bar{d}'$ such that $\forall C_i \in \mathscr{D}(\bar{d}, \bar{d}')$, $\mu_{R_i}(\bar{d}') > \min_{C_j \in \mathscr{D}(\bar{d}, \bar{d}')} \mu_{R_j}(\bar{d})$. Let C_j be the constraint for which this minimum holds.

$$\mu_{R_j}(\bar{d}') > \mu_{R_j}(\bar{d}).$$

Assume $\bar{d}$ is obtained by the algorithm and consider $\mathscr{C}_k$ such that $C_j \in \mathscr{C}_k$. To claim that $\bar{d}$ is not discrimin optimal is to claim that the set of constraints

$$\mu_{R_i}(\bar{x}) = \alpha_l \quad \forall C_i \in \mathscr{C}_l, l = 1, \ldots, k-1,$$

$$\mu_{R_j}(\bar{x}) > \alpha_k$$

is consistent (since $\bar{x} = \bar{d}'$ satisfies it) and this is a contradiction. □

The main features of the algorithm DA are its recursivity and its generality. It is recursive because in order to find the solutions with m fuzzy constraints, it finds all minimal sets of critical constraints and then the remaining ones form a problem to be solved in the same way at the next step. It requires the computation of the global satisfaction degree, the determination of the critical sets of constraints and the determination of their α-cuts.

In practice, computing critical sets of constraints can be very expensive. However, as shown in this report, there are classes of useful problems where the critical set is unique at each step and is rather easy to find.

4.3. Leximin-optimal algorithms

Algorithm LA

[breadth first] *Find the leximin solutions of* $\mathscr{P} = (\mathscr{X}, \mathscr{D}, \mathscr{C}, \mathscr{R})$

1. Compute the optimal satisfaction degree, i.e. the consistency degree

$$\alpha^* = \mathrm{Cons}(\mathscr{P}) = \sup_{\bar{d} \in \Omega} \min_{C_i \in \mathscr{C}} \mu_{R_i}(\bar{d}).$$

2. Determine the minimal subsets of critical constraints $\mathscr{C}_k^*$ (minimal according to the set-cardinality).
3. For every minimal set $\mathscr{C}_k^*$.

Find the leximin solutions of $\mathscr{P}' = (\mathscr{X}, \mathscr{D}, \mathscr{C}'_k, \mathscr{R}'_k)$ where

- each constraint $C_i \in \mathscr{C}_k^*$ has been transformed in the classical constraint $C'_i \in \mathscr{C}'_k$ associated with the α^*-section:

$$R'_i = R_{i_\alpha}^{=},$$

- each constraint $C_i \notin \mathscr{C}_k^*$ is kept with the restriction that its satisfaction degree has to be greater than α^*:

$$R'_i = R_i \cap R_{i_\alpha}^{>}.$$

4. Perform a pairwise leximin-comparison and delete leximin-dominated solutions.

As it can be seen, only a few things have been modified from DA to LA.

First, the notion of "minimal subset" has to be considered with respect to the set-cardinality. This difference between the algorithms coincides with the difference between the orderings, since Discrimin is based on set-inclusion and leximin relies on set-cardinality.

The second modification is more essential: not every solution built by the algorithm is optimal. As a matter of fact, every minimal critical subset found by the algorithm leads to complete solutions. In general, two minimal critical subsets found at the same step may lead to different satisfaction degree at the next step. Therefore, even if the minimal critical subsets are equally preferred, the solutions built on them may be different and since the leximin ordering is complete, they are comparable (contrary to the case of discrimin algorithm). That's the reason why at the end of the procedure, a pairwise leximin-comparison has to be done to eliminate the possible leximin-dominated solutions.

In fact, if the implementation of LA is iterative and involves a list of partial solutions being built, one can progressively perform the leximin-comparison and discard the dominated partial solutions.

It is possible to envisage a depth first version of the algorithm for finding the leximin optimal solutions because the leximin relation is a complete preordering.

Algorithm LA

[depth first]

- *Initialization*: Set the current satisfaction vector $CSV = 0$. The solution set is empty: $\mathscr{S} = \emptyset$. And the problem list contains only the problem $CSP = \{\mathscr{P}\}$.
- While the CSP-list is not empty,
  1. Select $\mathscr{P}^*$ the first element with respect to a L.I.F.O. strategy.
  2. Compute α^* the consistency degree of $\mathscr{P}^*$.
  3. Find the minimal critical subsets $\mathscr{C}_k^*$ and the corresponding partial vectors of satisfaction levels (minimal with respect to the set-cardinality).
  4. Compare each partial vector with the current satisfaction vector. If the result is O.K.,
     if there are still flexible constraints, update the CSP-list;
     if there is no more flexible constraints, update the $\mathscr{S}$-list.

The updating of the CSP-list consists in adding as many CSPs as there exists different minimal critical subsets. For every $\mathscr{C}_k^*$, the CSP to add is determined as previously: critical constraints are restricted to their α-section while other constraints are forced to reach a strictly higher satisfaction degree.

The updating of the $\mathscr{S}$-list is not a simple addition of new solutions to the previous set of solutions. Dominance checking can be performed, in order to delete solutions dominated with respect to the leximin ordering.

Finally, we have to precisely define how is computed the result of the comparison between a new partial solution (characterized by its partial satisfaction vector NSV) and the current satisfaction vector CSV. Both vectors are sorted in increasing order. Let j be the length of NSV. If CSV is shorter than NSV, then it is artificially completed with 0's on the right.

1. Compare component to component from $i = 1$ to $i = j$. If $CSV_i < NSV_i$, then CSV is set to NSV, the $\mathscr{S}$-list is emptied and the result is OK. STOP. If $CSV_i > NSV_i$, then the result is NOT OK. STOP.
2. Otherwise, the two vectors can not yet be distinguished. Set the result to OK. STOP.

The above algorithms lead to representing sets of discrimin or leximin optimal solutions under the form of non-fuzzy CSP problems. Namely for each fuzzy constraints C_i a threshold α_i has been found such that any solution to the classical CSP defined by $\{R^{=}_{i_{\alpha_i}}, i = 1, \ldots, n\}$ is discrimin (or leximin) optimal. So the above procedures are kind of defuzzification algorithms, that are concise description of the discrimin and leximin-optimal solutions.

Because of the DA algorithm and its properties (Theorem 2), it can easily be proved that a discrimin-optimal [5] solution can always be obtained by successive enhancements of any min-optimal solution. In general, several different min-optimal solutions may lead to the same discrimin-optimal solution; and, distinct discrimin-optimal solutions may sometimes be built from a given min-optimal one.

The behaviour of the leximin procedure is rather different, since from one min-optimal solution, it may happen that no leximin-optimal one is reachable. In other words, not all minimal critical subsets can be used to build a leximin-optimal solution.

Therefore, the two proposed algorithms (DA and LA) differ not only in the meaning of "minimal subset" (inclusion or cardinality based), but, if only one optimal solution is looked for, the required work is completely different.

[5] Recall that discrimin-maximal should be more adequate than discrimin-optimal, which is used for the sake of simplicity.

Assume, for example, an interactive framework, where the DM would like to contribute to the solution building. If the DM is convinced by the Discrimin ordering, each time the procedure proposes a set of "critical subsets", he can choose which one has to be further investigated. And so on, at each step. This interactive procedure requires much less effort than the complete DA, since only one $\mathscr{C}_k^*$ is checked (the interesting one!) and nevertheless, it gives finally an optimal solution.

On the other hand, it seems that no efficient leximin procedure can be interactive, since from a DM chosen minimal cardinality subset, it may be impossible to build a leximin-optimal solution.

5. Illustration on a flexible assignment problem

Before going into further details for particular cases, we treat now a combinatorial problem as a sample problem for our procedures. This problem belongs to a more general class of Boolean problems where the procedures can be used.

Let $E = \{e_1, \dots, e_m\}$ be a set of binary decisions. A solution F is a subset of E of a given size n that satisfies some constraints. Let $\mathscr{E}$ be the set of feasible solutions, supposed to be of the following form:

$$F \in \mathscr{E} \Rightarrow F \subseteq E \quad \text{and} \quad |F| = n.$$

Combinatorial problems of this type are, for example, the assignment problem (e_j is an assignment of a given operator to a given task), the travelling salesman (e_j can be the directed path between two cities). Each element e_i of E is characterized by a given satisfaction degree μ_i. Our aim is to find the solutions which maximize

$$\max_{F \in \mathscr{E}} \min_{e_i \in F} \mu_i$$

and are discrimin or leximin optimal.

The results of this section are applicable to all these problems. But, for simplicity, we focus on an assignment problem.

Each year, the army has to assign officers to tasks, according to their capabilities. Therefore, the candidates pass through several exams and a profile procedure gives for each candidate its degree of fulfillment of task requirements. Our problem is to find the best assignment of each officer to a task.

Let's take an example, where the candidates are not too good (sic.). The satisfaction matrix is as follows:

$$\begin{pmatrix} 2 & 1 & 1 & 1 & 6 \\ 1 & 3 & 3 & 5 & 0 \\ 2 & 1 & 4 & 0 & 1 \\ 1 & 3 & 2 & 4 & 0 \\ 2 & 0 & 1 & 1 & 5 \end{pmatrix}. \tag{41}$$

The satisfaction degrees have been given on a discrete scale between 1 and … 7. A row corresponds to an officer, while a column refers to a task. Even if nobody seems to be particularly able to cope with task 1, someone has to be assigned. Each task is as important as the others.

The decision set $\mathscr{X}$ consists in the task x_i assigned to each officer i: $\mathscr{X} = \{x_1, \dots, x_5\}$. The domain range of each officer is the set of tasks: $\forall i, x_i \in \{1, \dots, 5\}$. The rows in the assignment table are considered as vectors of satisfaction levels: if the first officer is committed to the last task, the related satisfaction degree is equal to 6.

A solution assigns a different task to each officer. Therefore, the solutions are characterized by the following relation:

$$\forall j \in \{1, \dots, 5\}, \exists! i{:}\, x_i = j.$$

Let's now build the discrimin and leximin optimal solutions of this problem.

5.1. Discrimin optimal solutions

1. Compute the global satisfaction degree:

$$\mathrm{Cons}(\mathscr{P}) = 2.$$

This can be easily obtained by considering the problem characterized by cancelling assignments with satisfaction less than 2 in the following matrix:

$$\begin{pmatrix} 2 & - & - & - & 6 \\ - & 3 & 3 & 5 & - \\ 2 & - & 4 & - & - \\ - & 3 & 2 & 4 & - \\ 2 & - & - & - & 5 \end{pmatrix}, \tag{42}$$

where – denotes infeasibility. This problem still has solutions, whilst removing the 2's implies a conflict between officers 1 and 5 on task 5.

2. To identify the critical subsets of constraints, we check the different combinations of officers satisfied with degree 2.

- Checking the subset $\{1\}$ as a critical subset comes down to verify the feasibility of the problem where the first officer is satisfied with degree 2 and all others are strictly more satisfied. In other words, does the following problem lead to a feasible solution?

$$\begin{pmatrix} 2 & - & - & - & - \\ - & 3 & 3 & 5 & - \\ - & - & 4 & - & - \\ - & 3 & - & 4 & - \\ - & - & - & - & 5 \end{pmatrix}. \tag{43}$$

  The answer is yes.

- The same test is performed for the singletons $\{3\}$ and $\{4\}$, without success. As a matter of fact, in both cases, two officers would again be assigned to the last task. Matrix (44a) (resp. (44b)) is related to the singleton $\{3\}$ (resp. $\{4\}$):

$$\begin{pmatrix} - & - & - & - & 6 \\ - & 3 & 3 & 5 & - \\ 2 & - & - & - & - \\ - & 3 & - & 4 & - \\ - & - & - & - & 5 \end{pmatrix}, \tag{44a}$$

$$\begin{pmatrix} - & - & - & - & 6 \\ - & 3 & 3 & 5 & - \\ - & - & 4 & - & - \\ - & - & 2 & - & - \\ - & - & - & - & 5 \end{pmatrix}, \tag{44b}$$

- The last singleton to check is $\{5\}$ which can lead to a feasible solution, since there exists at least one feasible solution to the following problem:

$$\begin{pmatrix} - & - & - & - & 6 \\ - & 3 & 3 & 5 & - \\ - & - & 4 & - & - \\ - & 3 & - & 4 & - \\ 2 & - & - & - & - \end{pmatrix}. \tag{45}$$

- Since neither 3 nor 4 belong to critical subsets of constraints, one can check the pair $\{3,4\}$ as a critical subset. But, the result is negative. No feasible solution can be obtained, if both officers 3 and 4 have to be satisfied with a degree equal to 2. It is illustrated by the following matrix:

$$\begin{pmatrix} - & - & - & - & 6 \\ - & 3 & 3 & 5 & - \\ 2 & - & - & - & - \\ - & - & 2 & - & - \\ - & - & - & - & 5 \end{pmatrix}. \tag{46}$$

  In other words, the critical subsets of constraints of $\mathscr{P}$ are $\{1\}$ and $\{5\}$.

3. For each identified critical subset, solve the reduced problem.

- Consider the problem, where the first officer is satisfied with a degree equal to 2. We need now to solve the following sub-problem $\mathscr{P}'$:

$$\begin{pmatrix} \infty & - & - & - & - \\ - & 3 & 3 & 5 & - \\ - & - & 4 & - & - \\ - & 3 & - & 4 & - \\ - & - & - & - & 5 \end{pmatrix}. \tag{47}$$

  The assignment which gives the first officer a satisfaction degree equal to 2 is now frozen. Therefore, it receives an ∞ degree.

  (a) The global satisfaction degree of $\mathscr{P}'$ is obviously equal to 3, since it is enforced by the second task.

  (b) We consider now the possible critical subsets of $\mathscr{P}'$. Both subsets $\{2\}$ (see Matrix (48a) and $\{4\}$ (see Matrix (48b) are critical ones, since they lead to a feasible solutions.

$$\begin{pmatrix} \infty & - & - & - & - \\ - & 3 & 3 & - & - \\ - & - & 4 & - & - \\ - & - & - & 4 & - \\ - & - & - & - & 5 \end{pmatrix}, \tag{48a}$$

$$\begin{pmatrix} \infty & - & - & - & - \\ - & - & - & 5 & - \\ - & - & 4 & - & - \\ - & 3 & - & - & - \\ - & - & - & - & 5 \end{pmatrix}. \tag{48b}$$

(c) The reduced problem corresponding to each of these two critical singletons has to be solved, iteratively. Since the sequel of the algorithm is obvious, we only state these reduced problems and give the solutions. The reduced problems are:

$$\begin{pmatrix} \infty & - & - & - & - \\ - & \infty & \infty & - & - \\ - & - & 4 & - & - \\ - & - & - & 4 & - \\ - & - & - & - & 5 \end{pmatrix},$$
$$\begin{pmatrix} \infty & - & - & - & - \\ - & - & - & 5 & - \\ - & - & 4 & - & - \\ - & \infty & - & - & - \\ - & - & - & - & 5 \end{pmatrix}. \tag{49}$$

They both lead to one solution $\bar{x}$ characterized by its satisfaction vector $\bar{u}$.

$$\begin{aligned} \bar{x} &= (1,2,3,4,5), & \bar{x} &= (1,4,3,2,5), \\ \bar{u} &= (2,3,4,4,5), & \bar{u} &= (2,5,4,3,5). \end{aligned} \tag{50}$$

- For the second critical subset $\{5\}$ of $\mathscr{P}$, the same procedure provides the two following solutions:

$$\begin{aligned} \bar{x} &= (5,2,3,4,1), & \bar{x} &= (5,4,3,2,1), \\ \bar{u} &= (6,3,4,4,2), & \bar{u} &= (6,5,4,3,2). \end{aligned} \tag{51}$$

5.2. Leximin optimal solutions

The computation of the leximin optimal solutions is approximatively the same as for the discrimin ordering. The difference occurs when checking if {3,4} is a minimal subset of constraints. Since its size is 2 and there exists singleton subsets of constraints (namely {1} and {5}), the pair can be immediately rejected.

At the end of the algorithm, we perform pairwise comparisons between the solutions, to remove the leximin dominated ones. Finally, we keep the solution

$$\begin{aligned} \bar{x} &= (5,4,3,2,1), \\ \bar{u} &= (6,5,4,3,2), \\ \bar{u}_{[]} &= (2,3,4,5,6). \end{aligned}$$

The Depth First version of the leximin-optimal algorithm allows to avoid some computations. It builds the first solution, which determines the current vector of satisfaction $CSV = (2,3,4,5,5)$. This solution is stored in $\mathscr{S}$.

During the two first steps of the second solution computation, the new satisfaction vector NSV can not be distinguished from CSV. But at the third step, NSV can help to reject the solution in $\mathscr{S}$, since NSV will be strictly better than CSV. The second generated solution comes into $\mathscr{S}$ and CSV becomes $(2,3,4,5,5)$.

As to the third solution, as soon as the third step, its NSV which is equal to $(2,3,4,4,-)$ is dominated by CSV. Therefore, the computations for the third solution are stopped.

Finally, the fourth solution is as good as the solution recorded in $\mathscr{S}$, during the four first steps. Then, it appears better than the solution in $\mathscr{S}$, which can be discarded. The leximin-optimal solution is the fourth solution.

This small example shows how the algorithmic refinements proposed in this version of the leximin algorithm can avoid useless computations and speed up the procedure.

5.3. Revisiting the algorithms

Other optimization procedures have been developed by Burkard and Rendl (1991) to solve “lexicographic bottleneck” problems with combinatorial structure. Their first algorithm converts the leximin problem into a sum optimization problem, by an appropriate scaling of the satisfaction degrees. As the authors themselves claim, the numbers constructed in this approach explode for large number of different satisfaction degrees (k) or for large problem size (n). Namely, the largest number is $(n+1)^k$.

In their second approach, they iteratively build an optimal solution to their leximin problem, using Sum Optimization Problem alternatively for determining the next satisfaction degree or for computing the number of saturated constraints. But they still need the use of very large numbers, of an order of magnitude similar to 2^n.

Our procedures are able to compute all discrimin or leximin optimal solutions, and do not make use of such increasing values. But, they use a list of partial solutions. We present now some variations of both basic procedures.

5.3.1. Algorithmic issues

For some classical O.R. problems, maxmin (or bottleneck) algorithms have been developed (see e.g. Martello and Toth, 1987). It is therefore easy to obtain the global satisfaction degree α^* of a problem $\mathscr{P}$. On the other hand, the determination of the critical subsets is not so easy.

However, it is possible to obtain an interval bracketing the size of the critical subsets. If we consider the following problem:

$$y = \max_{\bar{x}} \sum_i v_i(x_i),$$
$$v_i(w) = \begin{cases} 1, & \text{if } \mu_i(w) > \alpha^*, \\ 0, & \text{if } \mu_i(w) = \alpha^*, \\ -\infty, & \text{otherwise,} \end{cases} \tag{52}$$

where α^* is the consistency degree of the flexible problem $\mathscr{P}$. y provides the maximum number of constraints that can be satisfied at a higher degree that α^* (recall that already identified constraints at lower levels have been moved to ∞). Therefore, the lower bound for the critical set size is $(n-y)$.

An upper bound is given by the optimization of the following problem, which gives the maximal number of elements at degree α^*:

$$y = \max_{\bar{x}} \sum_i v_i(x_i),$$
$$v(w) = \begin{cases} 0, & \text{if } \mu_i(w) > \alpha^*, \\ 1, & \text{if } \mu_i(w) = \alpha^*, \\ -\infty, & \text{otherwise.} \end{cases} \tag{53}$$

This helps avoiding to check all the possible critical subsets, since too small or too large ones can be avoided. To check the remaining possible critical subsets A, we can solve the following maxmin problem:

$$z = \max_{\bar{x}} \min_i v_i(x_i),$$
$$v_i(w) = \begin{cases} \infty, & \text{if } \mu_i(w) = \alpha^* \text{ and } i \in A, \\ \mu_i(w), & \text{if } \mu_i(w) > \alpha^*, \\ -\infty, & \text{otherwise.} \end{cases} \tag{54}$$

The choice of the parameters v ensures that the solution to this problem will not include elements with low degree that do not belong to the critical subsets chosen at the previous steps.

The solution of this problem can have three kinds of value z:

$$z = \begin{cases} -\infty, & \text{the critical subset is not feasible,} \\ \infty, & \text{a complete discrimin optimal solution is reached,} \\ \alpha, & \text{the next satisfaction degree is } \alpha. \end{cases} \tag{55}$$

As to the leximin algorithms, it is enough to compute the lower bound of the critical subset size, since we are looking for minimal subset with respect to set-cardinality. But, the rest remains valid.

5.3.2. Example

Let’s consider again how to obtain the discrimin optimal solutions of the assignment problem characterized in (41). This example illustrates the usefulness of the bounds to avoid considering some subsets, namely the pair {3, 4}.

The lower bound for the critical subset size of $\mathscr{P}$ is obtained by the maxsum optimization of the assignment problem defined by the following matrix:

$$\begin{pmatrix} 0 & - & - & - & 1 \\ - & 1 & 1 & 1 & - \\ 0 & - & 1 & - & - \\ - & 1 & 0 & 1 & - \\ 0 & - & - & - & 1 \end{pmatrix}. \qquad (56)$$

The result is 4. In other words, the lower bound is equal to $5 - 4 = 1$.

The upper bound is also given by a maxsum optimization. Consider the assignment problem whose cost matrix is defined as

$$\begin{pmatrix} 1 & - & - & - & 0 \\ - & 0 & 0 & 0 & - \\ 1 & - & 0 & - & - \\ - & 0 & 1 & 0 & - \\ 1 & - & - & - & 0 \end{pmatrix}. \qquad (57)$$

The result is equal to 1. Therefore, there doesn't exist a feasible solution satisfying both constraints 3 and 4 at level 2.

6. Problems with unique discrimin optimal solutions

The above proposed algorithms are general but computationally very expensive; the knowledge of some particular problem structures can speed them up. Fortunately, this enhancement occurs in well-known and often applied mathematical models, which turn out to be CSP with fuzzy domains. From now on, we consider continuous CSP with fuzzy variables domain in a numerical framework.

First, we will prove that the discrimin-optimal solution to a convex problem is essentially unique. By essentially unique, we mean unique except, perhaps, for variables with satisfaction degree equal to one. We think this is a major reason why previous works have only been concerned by leximin. In most practical cases, including Linear Programming (LP), the discrimin solution is also the leximin one, because of its uniqueness.

For the LP problems, the paper of Behringer (1981) received little attention, probably because randomly chosen test-problems have a unique min-optimal solution. But in real-world problems and especially in combinatorial search, degeneracies occur and the discrimin search makes sense. We will briefly present some further results on this topic. It can be noted that the fuzzy PERT problem can be cast in this class.

Another framework to be considered is the "Isotonic Programming" model, which can also be viewed as a generalization of fuzzy PERT problems. The uniqueness of the discrimin optimal solution has also been obtained for isotonic problems.

6.1. Convex programming

A classical set $\mathscr{D}$ is "convex" if and only if each linear combination of any 2 elements of $\mathscr{D}$ belongs to $\mathscr{D}$.

$$\mathscr{D} \text{ convex} \Longleftrightarrow \begin{cases} \forall \bar{x}, \bar{y} \in \mathscr{D}, \\ \forall \kappa \in [0,1], \\ \kappa\bar{x} + (1-\kappa)\bar{y} \in \mathscr{D}. \end{cases}$$

A flexible domain $\tilde{X}$ is "essentially strictly convex" (es-convex) if and only if it is defined by a fuzzy interval whose bounds,if any, are strictly convex on (0,1).

$$\tilde{X}\text{es-convex} \Longleftrightarrow$$
$$\begin{cases} \forall x_1, x_2\ (x_1 \neq x_2) \text{ such that } \mu_{\tilde{X}}(x_i) \in (0,1)\ (i = 1,2), \\ \forall \lambda \in (0,1), \\ \mu_{\tilde{X}}(\lambda x_1 + (1-\lambda)x_2) > \min\limits_{i=1,2} \mu_{\tilde{X}}(x_i). \end{cases}$$

It is natural to consider a Feasibility Problem with such kind of preferences on the variables: it allows more-or-less preferred values as well as typical ones (the core of the fuzzy interval). Flexible constraint domains express the DM's preferences about the possible values of the variables, mainly in three different ways: left-bounded, right-bounded and bounded domains.

For example, when the DM says the ready date of an operation should be *before October 1st*, it makes sense to use a fuzzy number like Fig. 12. And the related variable s should take a value maximizing the membership degree to this fuzzy number. In the sequel, this kind of fuzzy number will be called *left-bounded fuzzy interval*. In order to increase the satisfaction degree of this con-

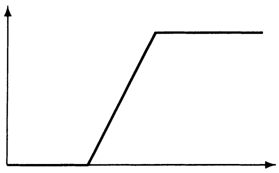

Fig. 12. $\tilde{X}$ is a left-bounded flexible domain.

straint we can only increase the value of the variable. This fuzzy number can also be called *increasing flexible domain* or *left-bounded flexible domain.*

The fuzzy intervals similar to the one of Fig. 13 will be called *right-bounded fuzzy interval.* They will be used to encode the preferences about the due dates. Finally, you can have also *bounded fuzzy intervals* resulting from the intersection of the two previous classes of domains. Each α-cut of such a domain is an interval.

Consider now the CSP with fuzzy domains ($\mathscr{P}$):

$$\begin{aligned} &\text{Maximize}\ {}_{\bar{x}}\min_{i\in I}\mu_{\tilde{X}_i}(x_i)\\ &\quad \bar{x}\in\mathscr{D}\subseteq\mathbb{R}^n, \end{aligned}\tag{58}$$

where $\mathscr{D}$ is convex and $\tilde{X}_i (i=1,\ldots,n)$ are es-convex.

Lemma 3. *If Eq. (58) is partially consistent, and if $\bar{x}$ and $\bar{y}$ are two min-optimal solutions to Eq. (58), then $\exists i\in I_F$ such that $x_i=y_i$* and *$\mu_{\tilde{X}_i}(x_i)=\mu_{\tilde{X}_i}(y_i)$ is the degree of consistency of the problem.*

Proof. Assume there exists 2 min-optimal solutions, $\bar{x}$ and $\bar{y}$ such that

$$\begin{aligned} &1>\mu_{\tilde{X}_k}(x_k)=\mu_{\tilde{X}_l}(y_l)=\text{Cons}(\mathscr{P})\\ &\quad\Rightarrow k\neq l \text{ or } x_k\neq y_l. \end{aligned}$$

In other words, we have $\forall i$

$$\begin{aligned} &\mu_{\tilde{X}_i}(x_i)=\mu_{\tilde{X}_i}(y_i)=\text{Cons}(\mathscr{P})\Rightarrow x_i\neq y_i,\\ &\mu_{\tilde{X}_i}(x_i)\neq\mu_{\tilde{X}_i}(y_i)\Rightarrow x_i\neq y_i. \end{aligned}$$

As a consequence of convexity of $\mathscr{D}$ and es-convexity of fuzzy domains, $\bar{z}=(\bar{x}+\bar{y})/2$ is min-better than any of $\bar{x}$ and $\bar{y}$. Indeed,

- either $\min(\mu_{\tilde{X}_i}(x_i),\mu_{\tilde{X}_i}(y_i))>\text{Cons}(\mathscr{P})$, and then,

$$\mu_{\tilde{X}_i}(z_i)\geqslant\min\left(\mu_{\tilde{X}_i}(x_i),\mu_{\tilde{X}_i}(y_i)\right)>\text{Cons}(\mathscr{P}),$$

- or $\min(\mu_{\tilde{X}_i}(x_i),\mu_{\tilde{X}_i}(y_i))=\text{Cons}(\mathscr{P})$ and then $x_i\neq y_i$ and

$$\mu_{\tilde{X}_i}(z_i)>\text{Cons}(\mathscr{P}).$$

Therefore, for all i, $\mu_{\tilde{X}_i}(z_i)>\text{Cons}(\mathscr{P})$, and $\bar{x}$ and $\bar{y}$ are not min-optimal solutions. □

An even stronger lemma can be proven, using the notion of critical subset. It claims that at the level $\alpha=\text{Cons}(P)$, the critical subset is unique. Let us note first that one critical subset is obviously non empty, otherwise $\alpha<\text{Cons}(P)$.

Lemma 4. *For any convex problem, like defined in Eq. (58), the critical subset at level* $\text{Cons}(\mathscr{P})=\alpha$ *is unique.*

Proof. Assume we have two disjoint critical subsets C and C'. It is possible to develop them into two different min-optimal solutions, $\bar{x}$ and $\bar{y}$. The solution $\bar{z}=(\bar{x}+\bar{y})/2$ is min-better than both $\bar{x}$ and $\bar{y}$, since $\forall i,\max(\mu_{\tilde{X}_i}(x_i),\mu_{\tilde{X}_i}(y_i))>\alpha$ and

$$\mu_{\tilde{X}_i}(z_i)>\min(\mu_{\tilde{X}_i}(x_i),\mu_{\tilde{X}_i}(y_i))\geqslant\alpha.$$

On the other hand, if we assume two different but non-disjunct critical subsets C and C', i.e. $C\neq C'$ and $C\cap C'\neq\emptyset$. In the worst case, the solution $\bar{z}$, built as previously as the mean of $\bar{x}$ and $\bar{y}$, has variables at the level α. These variables have to belong to $C\cap C'$ which is strictly smaller than both C and C'. Therefore, the critical subset related to $\bar{z}$ is included in C and C'. This is absurd, since C and C' have been assumed inclusion-minimal. □

Theorem 5. *The discrimin solution is essentially unique. Namely, all the discrimin solutions have all*

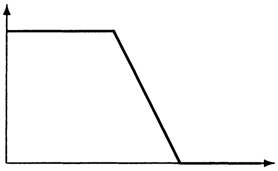

Fig. 13. $\tilde{R}$ is a right-bounded flexible domain.

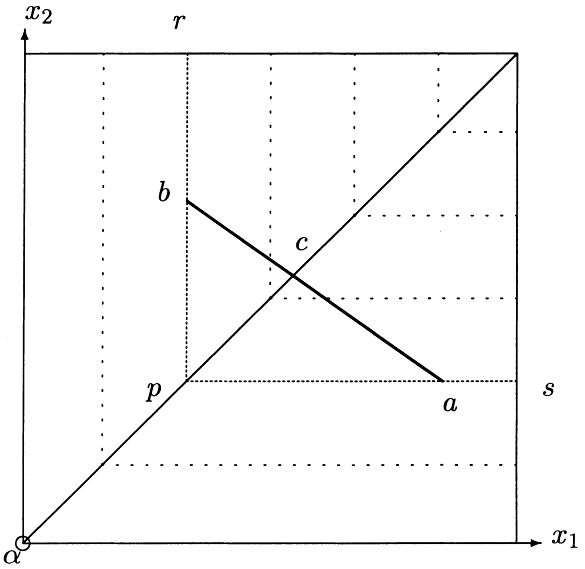

Fig. 14. Geometrical approach with linear membership functions on both variables.

their components equal except maybe those with satisfaction degree equal to one.

Proof. Assume it exists $\bar{x}$ and $\bar{y}$ two discrimin-optimal solutions, such that

$$\exists i\colon \min\{\mu_{\tilde{X}_i}(x_i), \mu_{\tilde{X}_i}(y_i)\} < 1 \text{ and } x_i \neq y_i.$$

Consider $\bar{z} = (\bar{x} + \bar{y})/2$ and compute $\mathscr{D}(\bar{x}, \bar{y})$. It is worth noticing that $x_i = y_i$ implies that $\tilde{X}_i \notin \mathscr{D}(\bar{x}, \bar{y})$.

- For all $\tilde{X}_l \in \mathscr{D}(\bar{x}, \bar{y})$, we have $x_l \neq y_l$ and

  $$\mu_{\tilde{X}_l}(z_l) > \min(\mu_{\tilde{X}_l}(x_l), \mu_{\tilde{X}_l}(y_l)).$$

- For all $\tilde{X}_l \notin \mathscr{D}(\bar{x}, \bar{y})$, such that $x_l \neq y_l$ and $\mu_{\tilde{X}_l}(x_l) = \mu_{\tilde{X}_l}(y_l) < 1$, we have $\mu_{\tilde{X}_l}(x_l) = \mu_{\tilde{X}_l}(y_l)$ and, again,

  $$\mu_{\tilde{X}_l}(z_l) > \min(\mu_{\tilde{X}_l}(x_l), \mu_{\tilde{X}_l}(y_l)).$$

Therefore, $\bar{z}$ should be discrimin-better than $\bar{x}$ and $\bar{y}$. In other words, $\bar{x}$ and $\bar{y}$ should not be discrimin-optimal. And the theorem is proved. □

The result is also an obvious consequence of Lemma 4 using the DA algorithm.

The previous theorems can be intuitively stated, for simple cases, on the basis of a geometrical approach.

Let us consider a problem in $\mathbb{R}^2$, where $\mathscr{D}$ is convex. The preferences on X_1 and X_2 are linear: $\mu_{\tilde{X}_i}(x_i) = x_i$. The global satisfaction degree α is equal to the minimum of x_1 and x_2. Instead of considering the problem in a 3-dimensions space, we project the two planes $\alpha = x_i$ onto the ground plane Ox_1x_2. We obtain the Fig. 14 where the dotted lines are iso-consistency lines.

Assume now that some solutions are min-optimal, with $\text{Cons}(\mathscr{P}) = 1/3$. These solutions have to be on the lines either (ps) or (pr). Should there be "min-optimal" solutions on both lines (e.g. a and b), there would exist at least a solution c better than the assumed min-optimal solutions, because the line between a and b belongs to the convex domain. Therefore, all the min-optimal solutions have to be on one line either (ps) or (pr) and they have in common a element: x_2 on (ps) or x_1 on (pr).

Similar reasoning holds when the membership functions are more general (e.g. see Fig. 15) or when the membership functions specify fuzzy numbers.

In the convex programming cases, the discrimin solution is essentially unique and can be obtained

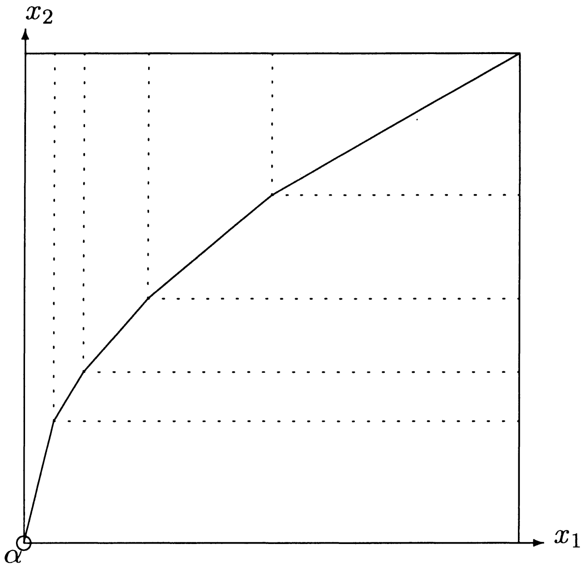

Fig. 15. Geometrical approach with linear (resp. quadratic) membership function for x_1 (resp. x_2).

step by step by our recursive procedure. At each step, the saturated variables are determined by the nonzero Lagrangian multipliers of the related $\alpha \leqslant \mu_{\tilde{X}_i}(x_i)$ constraints. And, for each saturated variable, the value is unique (because the solution is unique).

6.2. Fuzzy linear programming

Probably the best known model of fuzzy mathematical programming has been proposed by Zimmermann (1976, 1978). It has been first treated with triangular membership functions, but more general forms have been considered, e.g. in (Leberling, 1981).

In this linear programming model, aspirations levels for objective functions are given along with some tolerance, and (in)equalities in linear constraints are made flexible. The other parameters are crisp. The fuzzy objective functions and the fuzzy constraints receive the same treatment.

The non-fuzzy problem is of the form

$$\begin{aligned} &\text{find} && \bar{x}, \\ &\text{such that} && \bar{c}^T\bar{x} \geqslant b_0, \\ &&& (A\bar{x})_i \leqslant b_i \quad \forall i, \\ &&& \bar{x} \geqslant 0. \end{aligned} \tag{59}$$

Let us assume membership functions for the goal $\tilde{b}_0$ and the right-hand side coefficients $\tilde{b}_i$.

This problem can easily be cast in our model:

$$\begin{aligned} \max \quad & \text{“}\min\text{''}_{i=0,\ldots,n}\mu_{\tilde{b}_i}(\bar{y}_i) \\ \text{s.t.} \quad & \bar{c}^T\bar{x} \geqslant y_0, \\ & (A\bar{x})_i \leqslant y_i \quad \forall i, \\ & \bar{x} \geqslant 0. \end{aligned} \tag{60}$$

where the additional variables are $y_0, y_1, \ldots, y_n$.

It should be noted that the model of Zimmermann provides the first practical method to solve fuzzy linear programming, since it can be solved by crisp linear programming technique. In fact with linear membership functions, the fuzzy problem becomes (Zimmermann, 1976)

$$\begin{aligned} \max \quad & \alpha \\ \text{s.t.} \quad & \mu_0(\bar{x}) = 1 - \frac{b_0 - \bar{c}^T\bar{x}}{p_0} \geqslant \alpha, \\ & \mu_i(\bar{x}) = 1 - \frac{(A\bar{x})_i - b_i}{p_i} \geqslant \alpha \quad \forall i, \\ & \mu_0(\bar{x}), \mu_i(\bar{x}), \alpha \in [0,1]. \end{aligned} \tag{61}$$

As Linear Programming (LP) is one of the most visited domains of Mathematical Programming, Flexible Linear Programming (FLP) has received most attention devoted to Flexible Mathematical Programming. The simplicity and the power of the linear model are good reasons for this particular appeal, as well as the nice mathematical properties of this problem.

Behringer (1981) proposed a simplex based algorithm to solve a lexicographically extended linear maximin problem, which is very similar to our problem.

First, recall that FLP is included in Convex Programming and the theorem of the last section are also applicable here.

The first step of the discrimin algorithm consists of the computation of a maximin solution to the feasibility problem. Assuming preferences are expressed not on all variables, we obtain the general maximin problem:

$$\begin{aligned} &\text{Maximize } \lambda \\ &(C\ D)\begin{pmatrix}\bar{x}\\ \bar{y}\end{pmatrix} \leqslant \bar{b}, \\ &\mu_i(y_i) \geqslant \lambda \quad \forall i \in I_F, \\ &\bar{x},\bar{y},\lambda \geqslant 0, \end{aligned} \tag{62}$$

where $\bar{x} \in \mathbb{R}^k$, $\bar{y} \in \mathbb{R}^l$, $n = k + l$, $C \in \mathbb{R}^{m\times k}$, $D \in \mathbb{R}^{m\times l}$ and $\bar{b} \in \mathbb{R}^m$.

Let consider linear membership functions for left-bounded domains:

$$\mu_i(y_i) = \frac{y_i - p_i}{q_i}.$$

This gives the following vectors $\bar{p}, \bar{q} \in \mathbb{R}^l$.

$$\begin{aligned} &\text{Maximize } \lambda \\ &\begin{pmatrix} C & D & 0 \\ 0 & -I & \bar{q} \end{pmatrix}\begin{pmatrix}\bar{x}\\ \bar{y}\\ \lambda\end{pmatrix} \leqslant \begin{pmatrix}\bar{b}\\ \bar{p}\end{pmatrix} \\ &\bar{x},\bar{y},\lambda \geqslant 0, \end{aligned} \tag{63}$$

At the optimal solution of this maximin problem, the critical constraints will be characterized in the simplex tableau by a slack variable set to zero and a strictly positive marginal cost (Teghem, 1996). Let α^* be the optimal value of λ.

See (Behringer, 1981) for a more complex and complete procedure of selection. However, it isn't needed to compute the whole set of critical variables, if the remaining fuzzy domains are not modified. The whole set of critical variables will be determined in several simplex steps.

It is therefore very easy to identify critical variables. They are characterized by a critical constraint of the form $\mu_i(y_i) = (y_i - p_i)/q_i = \alpha^*$. Since the membership function is linear, the choice of the right value for the critical variable y_i is obvious. y_i is set to $p_i + q_i\alpha^*$.

This procedure can be applied iteratively to the problem and its successive reductions. It stops in a finite number of steps, since at each step the value of some variables is determined and the number of variables is finite.

6.3. Isotonic problems with fuzzy domains

A function is said to be *strictly isotonic* with respect to its arguments if and only if every increase in one of its arguments leads to an increase in the function value:

$$f(.) \text{ is strictly isotonic}$$

$$\Updownarrow$$

$$\forall u \geqslant u', v \geqslant v', \ldots \in \mathbb{R}$$

$$\Longleftrightarrow f(u,v,\ldots) > f(u',v',\ldots)$$

with at least, one strict inequality.

We will call *Isotonic Flexible Constraint Satisfaction Problem* (I-FCSP) every numerical problem enjoying the following characteristics. The set of variables of $\mathscr{X}$ is partitioned into two subsets $\mathscr{X}_1$ and $\mathscr{X}_2$ (i.e. $\mathscr{X}_1 \cap \mathscr{X}_2 = \emptyset$ and $\mathscr{X}_1 \cup \mathscr{X}_2 = \mathscr{X}$). On each variable of $\mathscr{X}_1$, there exists a left-bounded flexible domain on $\mathbb{R}$, noticing that high values are preferred for these variables. On the opposite, $\mathscr{X}_2$ contains the variables with right-bounded flexible domains, for which smaller values are preferred. Every crisp constraint CN_i requires that a particular (strictly isotonic) combination of some variables of $\mathscr{X}_1$ is smaller than a combination of a subset of $\mathscr{X}_2$.

In other words, the crisp constraints gave the following form:

$$f(x,y,z,\ldots) \leqslant g(r,s,t,\ldots).$$

where $f\colon \mathscr{Y} \subseteq \mathscr{X}_1 \to \mathbb{R}$ and $g\colon \mathscr{Z} \subseteq \mathscr{X}_2 \to \mathbb{R}$ are two strictly isotonic functions from subsets respectively of $\mathscr{X}_1$ and $\mathscr{X}_2$ to $\mathbb{R}$.

The class of isotonic problems differs from the class of convex problems as illustrated by the next example.

Example 2. The following small problem is non-convex but isotonic.

$$\max\ \min\{x/2, y/2\}$$
$$xy \leqslant 1,$$
$$x + y \leqslant 2.5.$$

The domain is not convex, since $\bar{a} = (2, 1/2)$ and $\bar{b} = (1/2, 2)$ belong to the domain, while $\kappa\bar{a} + (1-\kappa)\bar{b}$ doesn't for $\kappa \in (0,1)$. It is easy to check that the problem is isotonic. The optimal solution is $x = y = 1$.

6.3.1. Minimal I-FCSP problem

Consider a problem with a single crisp constraint and several fuzzy constraints, which we call *a minimal I-FCSP*.

Definition 8. The crisp constraint *CN* will be saturated at level α, if it holds with equality when all its variables are replaced by one of the extremal values of the α-cut of their flexible domain.

In a first lemma we prove that the global consistency degree of a I-FCSP problem $\mathscr{P}$ may always be viewed as the minimum of the consistency degrees of *minimal* I-FCSP problems $\mathscr{P}_i$ determined by every crisp constraint CN_i of $\mathscr{P}$. Therefore, solving a I-FCSP problem boils down to solving several successive minimal I-FCSP problems. This last task is considered in a second lemma.

Lemma 6 (Global consistency of a I-FCSP). *Consider a I-FCSP problem $\mathscr{P}$ as well as the set of minimal I-FCSP problems $\{\mathscr{P}_i\}_{i=0,\ldots,k}$ it defines. Namely, for each crisp constraint CN_i of $\mathscr{P}$, we build $\mathscr{P}_i$ with this unique crisp constraint as well as with the flexible domains over all variables of $\mathscr{X}$. We have then*

$$\mathrm{Cons}(\mathscr{P}) = \min_i \mathrm{Cons}(\mathscr{P}_i).$$

Proof. We have first that [6]

$$\begin{aligned}\mathrm{Cons}(\mathscr{P}) &= \sup_{\substack{\bar{x}\in\Omega \\ \forall i\in\{1,\ldots,k\}, f_i(\bar{x}\downarrow\mathscr{Y}_i)\leqslant g_i(\bar{x}\downarrow\mathscr{Z}_i)}} \min_{l\in\{1,\ldots,m\}} \mu_{\tilde{X}_k}(x_k) \\ &\leqslant \min_{i\in\{1,\ldots,k\}} \sup_{\substack{\bar{x}\in\Omega \\ f_i(\bar{x}\downarrow\mathscr{Y}_i)\leqslant g_i(\bar{x}\downarrow\mathscr{Z}_i)}} \min_{l\in\{1,\ldots,m\}} \mu_{\tilde{X}_k}(x_k) \\ &= \min_{i\in\{1,\ldots,k\}} \mathrm{Cons}(\mathscr{P}_i).\end{aligned}$$

Let α denote $\min_{i\in\{1,\ldots,k\}}\mathrm{Cons}(\mathscr{P}_i)$.

We consider now the vector $\bar{y} = (x_{1\alpha}, \ldots, x_{m\alpha})$ where e.g. $x_{i\alpha}$ is the extremal value of the α-level cut of $\tilde{X}_i$. If we prove that $\bar{y}$ is feasible with respect to all crisp constraints of $\mathscr{P}$, then

$$\mathrm{Cons}(\mathscr{P}) \geqslant \alpha = \min_{i\in\{1,\ldots,k\}} \mathrm{Cons}(\mathscr{P}_i)$$

which proves the lemma.

As a matter of fact, for every crisp constraint CN_i, we have $\alpha_i = \mathrm{Cons}(\mathscr{P}_i) \geqslant \alpha$ and therefore, $\bar{z}_i = (x_{1\alpha_i}, \ldots, x_{m\alpha_i})$ is feasible:

$$f_i(\bar{z}_i \downarrow \mathscr{Y}_i) \leqslant g_i(\bar{z}_i \downarrow \mathscr{Z}_i).$$

And because of the es-convexity of the flexible domains, $\bar{z}_i$ is unique.

We have that

$$\begin{aligned}&\forall X_k \in \mathscr{X}_1, \quad x_{k\alpha_i} \geqslant x_{k\alpha} \leftarrow f_i(\bar{y}\downarrow\mathscr{Y}_i) \leqslant f_i(\bar{z}_i\downarrow\mathscr{Y}_i),\\ &\forall X_k \in \mathscr{X}_2, \quad x_{k\alpha_i} \leqslant x_{k\alpha} \leftarrow g_i(\bar{y}\downarrow\mathscr{Z}_i) \geqslant g_i(\bar{z}_i\downarrow\mathscr{Z}_i),\end{aligned}$$

which leads to $\forall i$,

$$f_i(\bar{y}\downarrow\mathscr{Y}_i) \leqslant g_i(\bar{y}\downarrow\mathscr{Z}_i).$$

Hence $\bar{y}$ is feasible with respect to the crisp constraints of $\mathscr{P}$. □

[6] By $\bar{x}\downarrow\mathscr{Y}$, we denote the sub-vector of $\bar{x}$ corresponding to the subset of variables $\mathscr{Y}$.

In other words, the consistency degree of a I-FCSP problem is prescribed by solving each constraint independently from other constraints.

Let us now stress on the determination of the values of variables linked by a unique crisp constraint.

Lemma 7 (Choice of values for a saturated constraint). *Let a minimal I-FCSP problem* $\mathscr{P}$ *include the crisp constraint CN and the fuzzy domains* CF_i on $\mathbb{R}$ *for all the variables of CN. Let* $\mathrm{Cons}(\mathscr{P})$ *be the degree of consistency of the problem such that CN is saturated at level* $\alpha = \mathrm{Cons}(\mathscr{P})$. *Then the unique best choice of values is* $(x, y, w, \ldots, r, s, t, \ldots)$ *such that*

$$\mu_{\tilde{X}}(x) = \mu_{\tilde{Y}}(y) = \cdots = \mu_{\tilde{R}}(r) = \mu_{\tilde{S}}(s) = \cdots = \mathrm{Cons}(\mathscr{P})$$

The uniqueness holds if $\mathrm{Cons}(\mathscr{P}) < 1$.

Proof.

$$\begin{aligned}\mathrm{Cons}(\mathscr{P}) &= \max_{\substack{x,y,z,\ldots,r,s,t,\ldots \\ f(x,y,z,.) \leqslant g(r,s,t,.)}} \min\Big\{\mu_{\tilde{X}}(x), \mu_{\tilde{Y}}(y), \ldots, \\ &\qquad \mu_{\tilde{R}}(r), \mu_{\tilde{S}}(s), \ldots\Big\} = \max_{\substack{u,v \\ u \leqslant v}} \min\left\{\mu_{\tilde{F}}(u), \mu_{\tilde{G}}(v)\right\} \\ &= \Pi(\tilde{F} \leqslant \tilde{G}),\end{aligned}$$

where

$$\mu_{\tilde{F}}(u) = \max_{f(x,y,z,\ldots)=u} \min\left(\mu_{\tilde{X}}(x), \mu_{\tilde{Y}}(y), \ldots\right),$$

$$\mu_{\tilde{G}}(u) = \max_{g(r,t,s,\ldots)=u} \min\left(\mu_{\tilde{R}}(r), \mu_{\tilde{T}}(t), \ldots\right).$$

The calculation of $\Pi(\tilde{F} \leqslant \tilde{G})$ is pictured on Fig. 16.

Clearly, stating the FCSP comes down to using the extension principle of fuzzy arithmetic, and compute $\tilde{F} = f(\tilde{X}, \tilde{Y}, \tilde{Z}, \ldots)$ and $\tilde{G} = g(\tilde{R}, \tilde{S}, \tilde{T}, \ldots)$ using fuzzy arithmetic. This is easy because f and g are isotonic (Dubois and Prade, 1987).

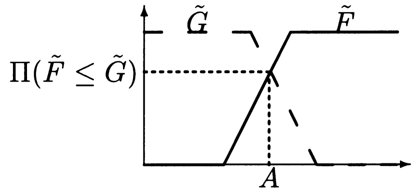

Fig. 16. $\tilde{F}, \tilde{G}$ and $\Pi(\tilde{F} \leqslant \tilde{G})$.

The value x such that $\mu_{\tilde{X}}(x) = \mathrm{Cons}(\mathscr{P})$ is unique, and also for $y, z, \ldots$ and $r, s, t, \ldots$, with identical satisfaction degree for all domains. And it is impossible to make a better choice.

As a matter of fact, if we want to increase one of the degrees, e.g. $\mu_{\tilde{X}}(x)$, we have to increase the value of x ($\tilde{X}$ is a left-bounded domain). But, this implies to reduce the value of y or z, or to increase the value of r, s or t, in order to respect the crisp constraint $f(.) \leqslant g(.)$. Whatever our choice for this, it will lead to a decrease in at least one of the membership values. Therefore, the global satisfaction degree will also decrease. □

When a problem is isotonic and the DA algorithm is applied to it, the above results tell that

- At each step the set of critical variables is unique. All variables involved in a minimal I-FCSP problem $\mathscr{P}_i$ such that $\mathrm{Cons}(\mathscr{P}_i) = \mathrm{Cons}(\mathscr{P})$ are critical.
- $\mathrm{Cons}(\mathscr{P})$ is easily obtained by solving each minimal I-FCSP $\mathscr{P}_i$ independently, and each such $\mathscr{P}_i$ is easily solved using fuzzy arithmetic.
- The defuzzification of critical variables gives unique values.

In summary, the relevant algorithm for I-FCSP problem is a particular instance of DA or LA. Since at each variable instantiation, there exist no choice for the value of the variables, this proves that the discrimin-optimal solution is unique and is also leximin-optimal.

6.3.2. Scheduling problem

The project scheduling problem can be described as a I-FCSP problem, although the classical representation of its constraints (28)–(30) does not fulfil the requirements of this particular framework.

As a matter of fact, the scheduling problem may be characterized by a set of crisp constraints pertaining to all the directed paths inside the problem graph. For every path $O_1, \ldots, O_k$, such that there exists a ready date s_1 for O_1 and a due date d_k for O_k (O_1 and O_k are not necessarily different), we may write

Table 1
Scheduling example

Task	Ready date	Due date	Operation	Duration	Machine
A	(0, 5, ∞, ∞)	(−∞, −∞, 20, 24)	a1	(4, 5, ∞, ∞)	1
			a2	(3, 4, ∞, ∞)	2
B	(0, 5, ∞, ∞)	(−∞, −∞, 20, 24)	b1	(2, 3, ∞, ∞)	2
			b2	(1, 2, ∞, ∞)	1
C	(0, 5, ∞, ∞)	(−∞, −∞, 24, 30)	c1	(2, 3, ∞, ∞)	1
			c2	(8, 9, ∞, ∞)	3
			c3	(4, 5, ∞, ∞)	2
D	(0, 5, ∞, ∞)	(−∞, −∞, 24, 30)	d1	(7, 8, ∞, ∞)	3
			d2	(8, 9, ∞, ∞)	1
E	(0, 5, ∞, ∞)	(−∞, −∞, 24, 30)	e1	(0, 1, ∞, ∞)	2
			e2	(6, 7, ∞, ∞)	3
			e3	(7, 8, ∞, ∞)	1
			e4	(2, 3, ∞, ∞)	3

$$s_1 + u_1 + u_2 + \cdots + u_k \leqslant d_k, \qquad (64)$$

where the starting time and the durations (u_i) belong to $\mathscr{X}_1$, the set of left-bounded variables, and the ending time belongs to $\mathscr{X}_2$.

In the problem solving, it is worth noticing that the saturated constraint of the form Eq. (64) precisely describes the so-called critical path. Therefore, the critical subset of flexible constraints contains fuzzy constraints on starting and ending times as well as durations.

For comparison, we study the example given in (Fargier, 1994). It is a Jobshop problem (5 tasks – 3 machines) (see Table 1). The ready dates, the durations and the due dates are flexible.

The capacity constraints have been transformed into precedence ones, by a Branch and Bound procedure. In this case, the procedure gives a sequencing for the operations (Table 2) which is min-optimal. This sequencing has to be instantiated, with respect to durations and starting times obeying the flexible constraints. We have therefore a flexible project scheduling to solve.

The precedence graph is depicted on Fig. 17.

The global satisfaction degree is $\frac{1}{3}$ and the critical path goes through d1 – e2 – c2 – c3. By the one-step constraint propagation, based on the core

Table 2
Given sequencing for the example

1	a1 – b2 – c1 – d2 – e3
2	b1 – e1 – a2 – c3
3	d1 – e2 – c2 – e4

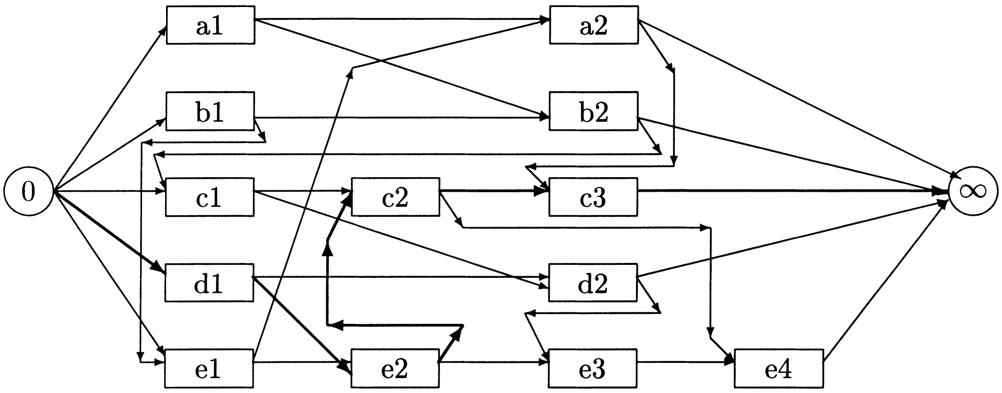

Fig. 17. The precedence graph related to the example from Fargier (1994). The critical path is drawn in bold.

of the slacks as proposed by (Fargier, 1994; Dubois et al., 1995) we obtain the following values:

Variable	Value	Satisfaction degree
Start of A	1.76	0.35
Start of B (*)	2.35	0.47
Start of C	7.47	1
Start of D	1.66	0.33
Start of E	5.75	1
End of A	10.37	1
End of B	7.47	1
End of C	28	0.33
End of D	18.1	1
End of E	27.8	0.35
Duration of a1	4.35	0.35
Duration of a2	4	1
Duration of b1 (*)	2.47	0.47
Duration of b2	1.36	0.35
Duration of c1	2.35	0.35
Duration of c2	8.3	0.33
Duration of c3	4.4	0.33
Duration of d1	7.34	0.33
Duration of d2	8.28	0.35
Duration of e1 (*)	0.62	0.62
Duration of e2	6.3	0.33
Duration of e3	10.4	0.35
Duration of e4	2.3	0.35

And the multi-step constraint propagation, as we propose, builds the following solution:

Variable	Value	Satisfaction degree
Start of A	1.76	0.35
Start of B (*)	3.4	0.68
Start of C	7.47	1
Start of D	1.66	0.33
Start of E	5.75	1
End of A	10.37	1
End of B	7.47	1
End of C	28	0.33
End of D	18.1	1
End of E	27.8	0.35
Duration of a1	4.35	0.35
Duration of a2	4	1
Duration of b1 (*)	2.68	0.68
Duration of b2	1.36	0.35
Duration of c1	2.35	0.35
Duration of c2	8.3	0.33
Duration of c3	4.4	0.33
Duration of d1	7.34	0.33
Duration of d2	8.28	0.35
Duration of e1 (*)	1	1
Duration of e2	6.3	0.33
Duration of e3	10.4	0.35
Duration of e4	2.3	0.35

The variables with an asterisk (*) have been enhanced.

7. Conclusion

The optimization problems with flexible constraints allow the Decision Maker to locally express his preferences on values of decision-variables. It should be stressed that we focus here on preference modelling and not on uncertainties. These new models are really powerful and they meet a practical demand. But they involve some difficult multicriteria issues. The classically used "conjunction order" involves a drowning effect and cannot always discriminate among solutions.

The discrimin order and the leximin order are more discriminating. But, some additional effort has to be done to obtain the optimal solutions with respect to these orders. We propose to apply a multi-step constraint propagation procedure. The method has been shown to give an optimal solution of both refined orders. Some uniqueness theorems have been proven for important subclasses of problems.

The theoretical framework developed in this paper is indeed wide enough to include a broad class of optimisation problems, obtained as a flexible generalization of the classical problems, such as flexible linear programming or flexible scheduling, for which the discrimin optimal solution is unique. However for discrete combinatorial problems, this uniqueness is far from warranted.

References

Behringer, F., 1977. On optimal decisions under complete ignorance: A new criterion stronger than both pareto and maxmin. European Journal of Operational Research 1, 295–306.

Behringer, F., 1981. A simplex based algorithm for the lexicographically extended linear maxmin problem. European Journal of Operational Research 7, 274–283.

Behringer, F., 1990. Lexmaxmin in fuzzy multiobjective decision making. Optimization 21, 23–49.

Bellman, R., Giertz, M., 1973. On the analytic formalism of the theory of fuzzy sets. Information Sciences 5, 149–156.

Bellman, R., Zadeh, L., 1970. Decision-making in a fuzzy environment. Management Science 17 (4), 141–164.

Benferhat, S., Cayrol, C., Dubois, D., Lang, J., Prade, H., 1993. Inconsistency management adn prioritized syntax-based entailment. Proceedings of the 13th Joint Conference on Artificial Intelligence (IJCAI), Chambery, pp. 640–645.

Brewka, G., 1989. Preferred subtheories: An extended logical framework for default reasoning. Proceedings of the 11th Joint Conference on Artificial Intelligence (IJCAI 89), Detroit, MI, pp. 1043–1048.

Burkard, R., Rendl, F., 1991. Lexicographic bottleneck problems. Operations Research Letters 10, 303–308.

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

Delgado, M., Verdegay, J., Vila, M., 1990. A possibilistic approach for multiobjective programming problem efficiency of solutions, in: Slowinski and Teghem (1990) pp. 229–248.

Descloux, J., 1963. Approximation in l^p and Chebyshev approximations. J. Soc. Indust. Appl. Math. 11, 1017–1026.

Dubois, D., Prade, H., 1978. Algorithmes de plus courts chemins pour traiter des données floues. RAIRO 12, 213–227.

Dubois, D., Prade, H., 1987. Théorie des Possibilités: applications à la représentation des connaissances en informatique, 2 ed., Masson, Paris.

Dubois, D., Fargier, H., Prade, H., 1994. Propagation and satisfaction of flexible constraints fuzzy sets neural networks and soft computing. Kluwer Academic Publishers, Dordrecht.

Dubois, D., Fargier, H., Prade, H., 1995. Fuzzy constraints in job-shop scheduling. Journal of Intelligent Manufacturing 6 (1995), 215–234.

Dubois, D., Fargier, H., Prade, H., 1996. Possibility theory in constraint satisfaction problems: Handling priority preference and uncertainty. Applied Intelligence 6, 287–309.

Dubois, D., Fargier, H., Fortemps, P., Prade, H., 1997. Leximin optimality and fuzzy set theroretic operations. IFSA'97.

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.

Erschler, J., Roubellat, J., Vernhes, 1976. Finding some essential characteristics of the feasible solutions for a scheduling problem. Operations Research 24, 774–783.

Esquirol, P., Lopez, P., Fargier, H., Schiex, T., 1995. Constraint programming. JORBEL 35, 5–36.

Fargier, H., 1994. Problèmes de satisfaction de contraintes flexibles: application à l'ordonnancement de production, Ph.D. Thesis, Université P. Sabatier Toulouse.

Fargier, H., 1997. Fuzzy scheduling: Principles and experiments. In: Dubois, D., Prade, H., Yagar, R.R. (Eds.), Fuzzy Information Engineering: A Guided Tour of Applications. Wiley, New York, pp. 655–668.

Fargier, H., Lang, J., Schiex, T., 1993. Selecting preferred solutions in fuzzy constraint satisfaction problems. Proc. of EUFIT'93 Aachen.

Fishburn, P.C., 1974. Lexicographic orders, utilities and decision rules: A survey. Management Science 20, 1442–1471.

Fortemps, P., 1997. Jobshop scheduling with fuzzy durations IEEE Transactions on Fuzzy Systems 5, 557–569.

Freuder, E., Snow, 1989. Improved relaxation and search methods for approximate constraint satisfaction with a maximin criterion. Proceedings of the 8th Conference of the Canadian Society for Computational Studies of Intelligence, pp. 227–230.

Hillier, F.S., Lieberman, G.J., 1989. Introduction to Operations Research, 4th ed. McGraw-Hill, New York.

Keeney, R., Raiffa, H., 1976. Decisions with Multiple Objectives: Preferences and value tradeoffs. Wiley, New York.

Leberling, H., 1981. On finidng the compromise solution in multicriteria problems using the fuzzy min operator. Fuzzy Sets and Systems 6, 105.

Lootsma, F.A., 1989. Stochastic and fuzzy PERT. European Journal of Operational Research 43, 174–183.

Mackworth, A., 1977. Consistency in networks of relations. Artificial Intelligence 8, 99–121.

Martello, S., Toth, P., 1987. Linear assignment problems. Annals of Discrete Mathematics 31, 259–282.

Montanari, H., 1974. Networks of constraints: Fundamental properties and applications to picture processing. Information Sciences 7, 95–142.

Moulin, H., 1988. Axioms of Cooperative Decision Making. Cambridge University Press, Cambridge, UK.

Pirlot, M., 1995. A characterization of 'min' as a procedure for exploiting valued preference relations and related results. Journal of Multi-Criterial Decision Analysis 4, 37–56.

Rice, J., 1962. Tschebyscheff approximation in a compact metric space. Bulletin of the American Mathematical Society 68, 405–410.

Sadeh, N., 1991. Look-ahead techniques for micro-opportunistic job shop scheduling. Report CS91-102, Carnegie Mellon University, Pittsburgh, PA.

Sakawa, M., 1993. Fuzzy Sets and Interactive Multiobjective Optimization. Plenum Press, New York.

Schweizer, B., Sklar, A., 1983. Probabilistic Metric Spaces. North-Holland, Amsterdam.

Sen, A.K., 1986. Social choice theory, In: Arrow, K.J., Intriligator, M.D., Handbook of Mathematical Economics 3, North-Holland, Amsterdam, pp. 1073–1181.

Slowiński, R., Teghem, J. (Eds.), 1990. Stochastic Versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Teghem, J., 1996. Programmation linéaire, collection statistique et mathématiques appliquées. Editions de l'Université de Bruxelles.

Van Hentenryck, P., 1989. Constraint Satisfaction and Logic Programming. MIT Press, Cambridge, MA.

Yager, R., 1988. On ordered weighted averaging aggregation operators in multicriteria decision making. IEEE Transactions on Systems, Man and Cybernetics 18, 183–190.

Zadeh, L.A., 1978. Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems 1, 3–28.

Zimmermann, H., 1976. Description and optimization of fuzzy system. International Journal of General System 2, 209–216.

Zimmermann, H., 1978. Fuzzy programming and linear programming with several objective functions. Fuzzy Sets and Systems 1, 45–55.