



Propagation and satisfaction of flexible constraints

Didier Dubois, Hélène Fargier, Henri Prade

► To cite this version:

Didier Dubois, Hélène Fargier, Henri Prade. Propagation and satisfaction of flexible constraints. Yager, Ronald R.; Zadeh, L.A. Fuzzy Sets, Neural Networks and Soft Computing, Chapter 7, Van Nostrand Reinhold; Wiley, pp.166–187, 1994, 978-0442016210. hal-04054177

HAL Id: hal-04054177

<https://hal.science/hal-04054177>

Submitted on 3 Apr 2023

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

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

7

Propagation and Satisfaction of Flexible Constraints

Didier Dubois, Hélène Fargier, and Henri Prade

Institut de Recherche en Informatique de Toulouse
Université Paul Sabatier
 118 route de Narbonne
 31062 Toulouse Cedex, France

Flexible constraints include both soft constraints (e.g., “the deadline is hopefully tomorrow, but the day after tomorrow is still acceptable”) and prioritized constraints (the idea is then to satisfy high priority constraints first and to look for the satisfaction of lower priority constraints only after, if possible). Possibility theory offers a convenient framework for modeling both types of constraints. This chapter shows how classical constraint satisfaction problem methods can be extended in a feasible way to flexible constraints. Moreover in flexible constraint satisfaction problems, solutions can be rank-ordered according to the preferences induced from the softness and the priorities.

7.1 INTRODUCTION

Many engineering problems such as design, planning, or scheduling may be modeled as constraint satisfaction problems (CSP) [26, 29]. A CSP is defined as finding any assignment of values to a set of variables ranging on, usually discrete, domains, so as to satisfy prescribed constraints. Most papers in this framework try to tackle the highly combinatorial nature (NP hard) of such problems, considering crisp constraints only [6, 28]. However, the classical CSP framework does not suit many practical situations, in which flexibility is an intrinsic characteristic of the information to be taken into account.

The notion of flexible constraint, whereby feasibility is a matter of degree, is not new [1] neither is the interest in fuzzy optimization problems (e.g., [21, 43]. Dubois [8] gives an example of a problem where fuzzy constraints are propagated in order to help the optimization procedure that determines design parameters;

see Wood and Antonsson [39] for another example. It is only recently that flexible constraints have become attractive in artificial intelligence [4, 16, 32]. Satoh [34] and Freuder [19] have stressed the flexible aspect of constraint satisfaction problems, and proposed theoretical frameworks for expressing prioritized constraints. Freuder [19, 20] developed general branch and bound techniques for dealing with these constraints without extending classical constraint propagation methods to them. Satoh [34] indicates that prioritized constraints modeled by logical formulas induce preferences among interpretations. Lang [24] handles prioritized constraints also, in the framework of possibilistic logic, a logic based on possibility theory, whose semantics rely on the notion of preference [14]. Other papers suggest that possibility theory [42] may be a suitable framework for the representation of prioritized constraints [35] and soft constraints [27]; moreover, they suggest new algorithms to handle such “fuzzy constraint satisfaction problems” (FCSP). Taking this a step further, we propose an approach, based on possibility theory, for representing and solving problems involving *both* prioritized constraints and soft constraints expressing preferences between values. We claim that both types of constraint can be regarded as local criteria inducing total orderings over tuples of values and can be represented by fuzzy restrictions [40].

The difficulty is then to compute the global ordering induced over solutions—and to choose the best one(s). Finding the (ordered) set of all the solutions becomes a problem of fuzzy restriction calculus. This problem is highly combinatorial when solved by global combination techniques, but local propagation algorithms may be applied when the problem presents a hypertree structure, as proposed by Pearl [30], Shafer and Shenoy [36], Dubois and Prade [12], Kruse and Schwecke [23] or Fonck and Straszcka [18]. We suggest that local propagation techniques can also be applied to more general hypergraphs in order to provide a good approximation to the solution. In the following we present efficient algorithms inspired by both Zadeh’s principle of combination-projection [41], and existing techniques in constraint propagation.

The next section deals with representation issues concerning soft constraints. Section 7.3 then explains how a CSP involving such constraints can be defined in the framework of possibility theory. In Sections 7.4 and 7.5, we present local computation schemes based on extensions of the CSP notions of arc consistency and path consistency, and propose an algorithm for computing the best solution(s). These techniques are illustrated in Section 7.6 by a simple example. Finally, nonmonotonic aspects of FCSPs are outlined in Section 7.7.

7.2 REPRESENTATION ISSUES

A hard or crisp constraint C between a set of k variables $X = \{x_1, \dots, x_k\}$ can be represented by means of a nonfuzzy relation R on $U_1 \times \dots \times U_k$, U_j being the range of variable x_j ; R is the crisp subset of $U_1 \times \dots \times U_k$ of k -tuples of values (u_1, \dots, u_k) (also called “instantiations” of $\{x_1, \dots, x_k\}$) that satisfy C . The set $\{x_1, \dots, x_k\}$ of variables involved in R is denoted $V(R)$.

7.2.1 Representing Soft Constraints

A soft constraint expresses preferences between k -tuples of values that can be assigned to a set of variables. In this chapter we assume that these preferences can be modeled by means of a total order and that this total order is encoded by means of a fuzzy relation R , that assigns to each k -tuple (u_1, \dots, u_k) its level of preference $\mu_R(u_1, \dots, u_k)$ in a totally ordered set L . This set is usually chosen as the unit interval $[0, 1]$. $\mu_R(u_1, \dots, u_k) > \mu_R(u'_1, \dots, u'_k)$ means that (u_1, \dots, u_k) is preferred to (u'_1, \dots, u'_k) as a value of (x_1, \dots, x_k) , and $\mu_R(u_1, \dots, u_k) = 0$ if (u_1, \dots, u_k) is a forbidden k -tuple. $\mu_R(u_1, \dots, u_k) = 1$ means that (u_1, \dots, u_k) totally satisfies the constraint. More generally, $\mu_R(u_1, \dots, u_k)$ can also be interpreted as degrees of satisfaction of the soft constraint C . $\mu_R(u_1, \dots, u_k) > 0$ corresponds to a feasible assignment, even if it only partially satisfies C . Usually, there exists at least one tuple (u_1, \dots, u_k) that totally satisfies the constraint C (feasibility of the constraint), i.e., $\exists(u_1, \dots, u_k), \mu_R(u_1, \dots, u_k) = 1$. R is then said to be normalized.

A fuzzy relation restricts the possible values that can be assigned to the variables in accordance to the preference criterion C . Hence the concept of a fuzzy restriction, that is "a fuzzy relation which acts as an elastic constraint on the values that may be assigned to variables" [40], seems to be suitable for capturing such information. In the terminology of possibility theory, μ_R acts as a possibility distribution on $U_1 \times U_2 \times \dots \times U_k$, which restricts the more or less possible values of the k -tuple of variables (x_1, \dots, x_k) , which are admissible according to C .

7.2.2 Representing Prioritized Constraints

Fuzzy restrictions also offer a suitable formalism for the expression of prioritized constraints. When it is possible to *a priori* exhibit a complete order over the respective priorities of the constraints, these priorities will be represented by means of levels in the scale $[0, 1]$: a coefficient α_C is attached to each constraint C (its priority degree) and indicates the degree to which C must be satisfied. If $\alpha_C = 1$, C is an imperative constraint; if $\alpha_C = 0$, it is completely possible to violate C (C has no importance in the problem). Given two constraints C and C' : $\alpha_C > \alpha_{C'}$ means that the satisfaction of C is more necessary than the satisfaction of C' ; if C and C' cannot be satisfied simultaneously, solutions compatible with C are preferable to solutions compatible with C' .

Priorities on constraints can be transformed, without any loss of information, into satisfaction degrees on values. Indeed, since α_C represents to what extent it is necessary to satisfy C , $1 - \alpha_C$ indicates to what extent it is possible to violate C . In other words, *any* instantiation (u_1, \dots, u_k) satisfies C to a degree greater than or equal to $1 - \alpha_C$. Hence, the pair (C, α_C) , where C is a crisp constraint, can be modeled as a soft constraint represented by the fuzzy relation S on $U_1 \times \dots \times U_k$ (see Fig. 7.1):

$$\begin{aligned} \mu_S(u_1, \dots, u_k) &= 1 && \text{if } (u_1, \dots, u_k) \text{ satisfies } C \\ &= 1 - \alpha_C && \text{if } (u_1, \dots, u_k) \text{ violates } C \end{aligned}$$

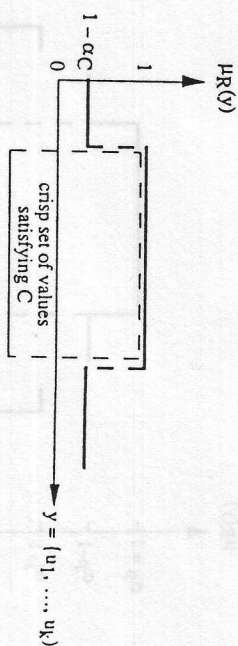


FIGURE 7.1. A crisp constraint C with priority α_C .

More generally, if C is a soft constraint modeled by the fuzzy relation R whose priority is α_C , (C, α_C) is represented by the fuzzy relation S :

$$\mu_S(u_1, \dots, u_k) = \max[1 - \alpha_C, \mu_R(u_1, \dots, u_k)]$$

These definitions are in accordance with the treatment of certainty-qualified assertions in possibility theory, and the principle of minimum specificity, interpreting priority levels as degrees of necessity [15]. Moreover they are, when R is crisp, in full agreement with possibilistic logic [19].

It should be noticed that a soft constraint C with preferences described by a fuzzy relation R involving a finite number of satisfaction degrees $0 = \alpha_0 < \alpha_1 < \dots < \alpha_{p-1} < \alpha_p = 1$ can be represented by a finite set of prioritized constraints $\{C_j^i, 0 \leq j < p\}$, following a remark in Prade, [31] (see Fig. 7.2):

$$\begin{aligned} \alpha_{C_j} &= 1 - \alpha_j \text{ and } R^j = \text{crisp set of } k\text{-tuples satisfying } C^j \\ &= \{(u_1, \dots, u_k), \mu_R(u_1, \dots, u_k) > \alpha_j\} \end{aligned}$$

7.2.3 Operations on Fuzzy Relations

Operations of conjunctive combination and projection can be extended to fuzzy relations [40, 41].

Given $W = \{x_{w1}, \dots, x_{wk}\}$ and $Y = \{x_{y1}, \dots, x_{yn}\}$ two subsets of $\{x_1, \dots, x_n\}$ such that $W \subseteq Y$ and a fuzzy relation T restricting the possible values of Y , the *projection* (or the *marginalization*) of T on W is a fuzzy relation $R = (T)^{W, W}$ restricting the possible values of W . It is defined on $U_{w1} \times \dots \times U_{wk}$ by

$$\mu_R(u_{w1}, \dots, u_{wk}) = \sup_{\{u_{y1}, \dots, u_{yn} \mid (u_{w1}, \dots, u_{wk}, u_{y1}, \dots, u_{yn}) \in T\}} \mu_T(u_{w1}, \dots, u_{wk}, u_{y1}, \dots, u_{yn})$$

where $(u_{y1}, \dots, u_{yn})^{W, W}$ denotes the restriction of (u_{y1}, \dots, u_{yn}) to W , $\mu_R(u_{w1}, \dots, u_{wk})$ estimates to what extent the instantiation (u_{w1}, \dots, u_{wk}) , which is a partial instantiation of Y , can be extended to a complete instantiation of Y that satisfies T .

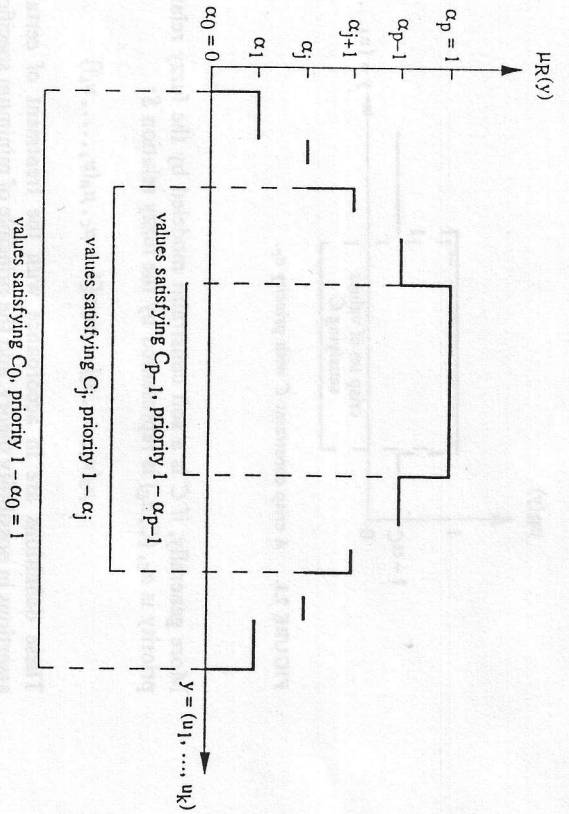


FIGURE 7.2. Decomposition of a soft constraint into a family of prioritized constraints.

The *conjunctive combination* of two fuzzy restrictions R and S restricting the possible values of two sets of variables X and Y is a fuzzy restriction over the possible values of $W = X \cup Y$. It is defined by $T = R \otimes S$:

$$\mu_T(u_{w,1}, \dots, u_{w,k}) = \min[\mu_R(u_{w,1}, \dots, u_{w,k})^{1^X}, \mu_S(u_{w,1}, \dots, u_{w,k})^{1^Y}]$$

$\mu_T(u_{w,1}, \dots, u_{w,k})$ estimates to what extent the variable instantiation $(u_{w,1}, \dots, u_{w,k})$ of W satisfies both R and S .

Note that the use of the combination rule underlies an assumption of *commensurability* between preference levels pertaining to different constraints. In other words the user who specifies the constraints must describe them by means of a unique preference scale L . Moreover, priority levels must be chosen on the dual scale $\{(c(\ell), \ell \in L)\}$ where c is an order-reversing mapping) so as to acknowledge the transformation of priority levels on constraints into preference degrees on solutions. Although natural and often implicit, these assumptions must be acknowledged when fuzzy set-based approaches are used.

Besides soft constraints differ from criteria (or objective functions) by the fact that no compensation of the extents to which they are satisfied is allowed among them, as modeled by the min operation (which is a purely logical conjunction operation that retains the least degree of satisfaction).

7.3 THE FUZZY CONSTRAINT SATISFACTION PROBLEM

An FCSP is defined by a set of m , possibly fuzzy (soft or prioritized), constraints $P = \{R_1, \dots, R_m\}$ restricting the possible values of n variables, say $\{x_1, \dots, x_n\}$, each x_i ranging on a domain U_i here supposed to be finite.

Given an FCSP $P = \{R_1, \dots, R_m\}$ restricting the possible values of $X = \{x_1, \dots, x_n\}$, the definition of conjunctive combination states that $\mu_{R_1 \otimes \dots \otimes R_m}(u_1, \dots, u_n)$ estimates to which extent the variable instantiation (u_1, \dots, u_n) satisfies all the constraints. That is to say, $\mu_{R_1 \otimes \dots \otimes R_m}(u_1, \dots, u_n)$ is the satisfaction degree of P by (u_1, \dots, u_n) . It is equal to the satisfaction degree of the constraint that is the least satisfied by (u_1, \dots, u_n) . Thus, the best solutions of P are the instantiations whose satisfaction degree of the least satisfied constraint is maximal. The combination of preference is noncompensatory for flexible constraints, as opposed to combination modes in multiobjective utility theory [22] where trade-offs are acceptable. The violation of one constraint is not supposed to be counterbalanced by the full satisfaction of other constraints.

$\mu_{R_1 \otimes \dots \otimes R_m}(u_1, \dots, u_n)$ is the membership degree of (u_1, \dots, u_n) to the set of all the solutions, which is the fuzzy set $\rho = R_1 \otimes \dots \otimes R_m$. The membership degrees to the fuzzy set of solutions discriminate the potential solutions since they induce a complete order over the instantiations; under the commensurability assumption, this order does not depend on whether L is a numerical scale. In other terms, the FCSP approach to flexibility is more qualitative than quantitative. Actually, solving a classical CSP means separating the set of all instantiations into two classes: the instantiations that are solutions to the problem and those that are not. Introducing flexibility just refines this ordering.

Conversely, the set of fuzzy restrictions $\{R_1, \dots, R_m\}$ can be regarded as a *decomposition* [40] or a *factorization* [36] of a global fuzzy relation $\rho = R_1 \otimes \dots \otimes R_m$ restricting the combinations of values that may be assigned to (x_1, \dots, x_n) .

Even if each constraint is generally normalized in practice, ρ may be sub-normalized if some constraints are conflicting, i.e., the problem can be partially or even totally inconsistent. The height of ρ is in fact the consistency degree of the FCSP:

$$\text{Cons}(P) = \text{Height}(\rho) = \sup_{(u_1, \dots, u_n) \in U_1 \times \dots \times U_n} \mu_\rho(u_1, \dots, u_n)$$

is the consistency degree of P . It is equal to the extent to which an instantiation exists that satisfies the set of constraints. If $\text{Cons}(P) < 1$, it is not possible to totally satisfy all the constraints. A best solution in this case implicitly involves the relaxation of some constraints. This relaxation is automatically done, and guided by the preference levels. By complementation we define the inconsistency degree of P

$$\text{Incons}(P) = 1 - \text{Height}(\rho) = 1 - \sup_{(u_1, \dots, u_n) \in U_1 \times \dots \times U_n} \mu_\rho(u_1, \dots, u_n)$$

7.3.1 Using Combination/Projection Principle to Make Induced Constraints Explicit

Since $R_1 \otimes \dots \otimes R_i \otimes \dots \otimes R_m \subseteq R_i$ (where \subseteq is the fuzzy set inclusion defined by the inequality \leq between the membership functions), the inclusion $\rho^{1(x_k)} \subseteq R_i$ holds, which means that R_i may not explicitly account for all the dependencies between the variables it relates. Indeed, further restrictions may be induced by other constraints. For instance, the problem $P = (R_1, R_2)$, where R_1 : " $x_1 = x_2$ " and R_2 : " $x_2 = x_3$," induces a restriction R_3 : " $x_1 = x_3$ " that is not expressed in P .

The induced constraints may be computed according to the combination/projection principle [40, 41]: the constraint R induced by a set of constraints $\{R_1, \dots, R_m\} \subseteq \{R_1, \dots, R_m\}$ on a set of variables $\{x_1, \dots, x_m\} \subseteq \{x_1, \dots, x_m\}$ is defined by $R = (R_1 \otimes \dots \otimes R_m)^{1(x_1, \dots, x_m)}$.

It should be pointed out that

$$(R_1 \otimes \dots \otimes R_m)^{1(x_1, \dots, x_m)} \otimes (R_1 \otimes \dots \otimes R_m) = R_1 \otimes \dots \otimes R_m$$

according to the properties of projection and combination, especially the idempotence of the combination. Hence, problems $P = \{R_1, \dots, R_m\}$ and $P' = \{R_1, \dots, R_m\} \cup \{R\}$ have the same fuzzy set of solutions. They are said to be *equivalent*.

In other words, there may be several decompositions of the same fuzzy relation ρ corresponding to equivalent FCSP in which the constraints are more or less explicit. A problem can be transformed into an equivalent one by making induced constraints explicit.

7.3.2 Interactivity and Separability

Even if not stated explicitly in $\{R_1, \dots, R_m\}$, ρ also induces a fuzzy relation on each variable x_1, \dots, x_m : $\rho^{1(x_i)}$ is the (explicit) restriction of possible values that can be assigned to x_i in accordance with P no matter what values are assigned to the other variables. Most of the time values that can be assigned to one variable are not independent of the values that can be assigned to the others, i.e., $\rho^{1(x_i, x_j)} \subset \rho^{1(x_i)} \otimes \rho^{1(x_j)}$.

In case of equality, both variables are said to be noninteractive [41]. If all the variables are noninteractive, $\rho = \bigotimes_{i=1, \dots, m} \rho^{1(x_i)}$ and ρ is said to be *separable*. FCSP are seldom separable since constraints express dependencies at least between pairs (or more generally k -tuples) of variables. In the general case we have:

$$\rho \subseteq \bigotimes_{i=1, \dots, m} \rho^{1(x_i)}$$

The concept of separability can be generalized using the following definition of k -decomposability: a relation ρ is k -decomposable if it can be decomposed into a set of relations $\{R_1, \dots, R_m\}$ such as $\rho = \bigotimes_{i=1, \dots, m} R_i$ and $\text{Card}[V(R_i)] \leq k$, $i = 1, m$. One-decomposability is equivalent to separability.

7.3.3 Fuzzy Constraint Satisfaction Problems

In summary, an FCSP is defined by a set of m fuzzy relations $P = \{R_1, \dots, R_m\}$ restricting the possible values of n variables $\{x_1, \dots, x_n\}$, and represents a global restriction $\rho = \bigotimes_{i=1, \dots, m} R_i$ on the values that can be assigned to the variables in accordance with P . The following classical problems in the constraint satisfaction framework have equivalent counterparts in the flexible setting:

- Determine the existence of a solution $\Leftrightarrow \text{compute } \text{Cons}(P) = \text{Height}(\rho)$.
- Find one of the best solutions \Leftrightarrow determine any instantiation (u_1, \dots, u_n) such that $\mu_\rho(u_1, \dots, u_n) = \text{Cons}(P)$.
- Determine the relation restricting ρ to compute $\rho^{1(x_1, \dots, x_m)}$ $\{x_1, \dots, x_m\}$ the values of $\{x_1, \dots, x_m\}$.
- Find all the possible values for variable $x_i \Leftrightarrow$ compute $\rho^{1(x_i)}$.

Hence FCSP reduce to fuzzy restriction calculus problems.

7.3.4 FCSP vs. Other Approaches to Flexibility in CSP

Freuder [19] and Satoh [34] devised theoretical foundations for the treatment of flexibility in CSPs. Satoh tries to apply results on circumscription to the handling of prioritized constraints so as to induce preference relations on the solution set. A similar point of view is adopted in Lang [24] where prioritized constraints are expressed in possibilistic logic. Although in accordance with ours, Satoh's approach [34] differs in the way priorities between constraints are expressed. Indeed, Satoh uses second-order logic to describe priorities. Moreover the ordering of solutions depends on how many constraints are satisfied. In our approach, solutions that satisfy an FCSP to the same degree are not discriminated, even if some of them satisfy more constraints. In other terms, the best solutions in the sense of Satoh are among the best according to the FCSP definition. However, a lexicographic ordering may be used in FCSP, if needed, to discriminate solutions sharing the same global satisfaction degree, as proposed by Descottes and Latombe [7]; the idea is to compare the vectors of the components of the evaluation rather than the scalar values resulting of the aggregation of the component values of each vector.

Taking a dual point of view, Freuder [19] regards a flexible problem as a collection of classical CSPs. A metric can then be defined that evaluates the distance between them. Then, the question is to "find the solutions to the closest solvable problem." The FCSP approach is perfectly in accordance with this view of constraint relaxation by partial satisfaction, as soon as a weight is associated with each possible weakening of each constraint C_i and the metric defined by the maximum of the weights of the relaxations performed. Indeed, an FCSP involving p different satisfaction levels is equivalent to p CSPs: for each level $\alpha_j > 0$, $\alpha_j \in L$, a CSP P^{α_j} is constituted by the set of hard constraints $C_i^{\alpha_j}$ containing the tuples that satisfy C_i to a degree greater than or equal to α_j .

The set of best solutions to the flexible problem is the set of solutions of the consistent P^2 -of higher α_j (the closest solvable problem in the sense of Freuder),

Associating priorities to constraints has also been proposed [3, 7, 17, 35]. The aim is still to minimize the maximum of the priority levels of the unsatisfied constraints. Modeling soft constraints by means of fuzzy sets is a more recent proposal [4, 5, 27, 32]; besides Faltings et al. [16] present an approach to dynamic CSP where they distinguish between a set of feasible values and a set of preferred values, which is a rough version of FCSP. In fact, FCSP is nothing but a generalization of these two types of approaches (priorities as well as soft constraints).

Moreover, the FCSP approach departs from probabilistic or cost-based approaches, where the best solutions are those for which the sum of satisfaction degrees is maximal [33]. In these approaches the violation of a constraint can be counterbalanced by the satisfaction of other constraints. In an FCSP, as soon as an instantiation violates a hard constraint, it is totally inconsistent:

$$\mu_{R_1} \otimes \cdots \otimes \mu_{R_m}(u_1, \dots, u_n) = 0$$

Thus, we are in accordance with the following constraint satisfaction principle: no constraint can be violated—except in the limits of its relaxation possibilities, which are expressed in the FCSP formalism by means of soft constraints.

7.4 LOCAL COMPUTATION SCHEMES

Computing $\rho = \bigotimes_{i=1, \dots, m} R_i$ or some of its projections is an NP-hard problem when using a global combination technique. These tasks can be approximated using local computation techniques, whose aim is to transform the problem into an equivalent one in which some induced constraints have been made explicit (in the best case, they are equal to the appropriate projection of ρ).

To graphically represent the structure of the problem, a hypergraph is often defined whose nodes are the variables $\{x_1, \dots, x_n\}$ and hypertedges represent constraints R_i relating a subset of variables. An unary restriction A_j is supposed to be associated with each variable x_j : by default, A_j stands for “ $x_j \in U_j$ ” (no unary constraint). A_j is an upper approximation of the fuzzy set of possible values of x_j according to the global problem: $A_j \supseteq \rho^{(1,*)}$. The two following local computation schemes will improve this approximation.

7.4.1 Making Unary Restrictions Explicit by

Ensuring Arc Consistency

The fuzzy unary restriction A_j of the possible values of a variable x_j is said to be consistent with the hypereedge R_i relating x_j to variables $V(R_i) - \{x_j\} = \{x_1, \dots, x_k\} - \{x_j\}$ if: $A_j \subseteq ([R_i \otimes (\bigotimes_{l=1, \dots, k, l \neq j} A_l])^{(x)})^{(x)}$ where \subseteq is the fuzzy set inclusion.

An FCSP is said to be arc consistent if, for each variable x_j , the associated unary restriction is consistent with all the restrictions relating this variable to some others, formally, if $\forall x_j, VR_i \in \{R_1, \dots, R_m\}$ s.t. $x_j \in V(R_i) = \{x_1, \dots, x_k\}$,

$$A_j \subseteq \left[R_i \otimes \left(\bigotimes_{l=1, \dots, k, l \neq j} A_l \right) \right]^{\uparrow(x_j)} \quad (1)$$

In other terms, a problem is arc consistent if the neighbors of each x_j do not induce on x_j other restrictions than those already described by the unary constraint A_j associated with x_j , i.e., if these induced restrictions are explicit in A_j . We will define a new local propagation scheme that transforms a fuzzy CSP into an equivalent one verifying the property of arc consistency.

Arc consistency of the FSCP is ensured by changing all A_j s restricting variables x_j into $A'_j = A_j \otimes R_{i \rightarrow x_j}$ for each R_i iteratively, where $R_{i \rightarrow x_j} = [R_i \otimes (\bigotimes_{k \in V(R_i), k \neq j} A_k)]^{\perp(x_j)}$. Note that $A'_j = [R_i \otimes (\bigotimes_{k \in V(R_i), k \neq j} A_k)]^{\perp(x_j)}$. Since making a constraint explicit does not change the set of solutions, we can update the problem by replacing A_j by A'_j . The modified problem is equivalent to the previous one, but x_j is now consistent with the hyperedge R_i . After being updated by arc consistency, A'_j is a better approximation of $\rho^{\perp(x_j)}$ than A_j .

For instance, let A_1 (resp. A_2) be a fuzzy set restricting the values of x_1 (resp. x_2) and R models the fuzzy rule “if x_1 is C , then x_2 is D ”: $\mu_R(u_1, u_2) = \mu_C(x_1) \rightarrow \mu_D(x_2)$ for some implication function \rightarrow . The restriction induced via arc consistency on x_2 is $B' = (R \otimes A_1 \otimes A_2)^{1(x_2)}$; clearly, $(R \otimes A_1)^{1(x_2)}$ computes the generalized modus ponens with fuzzy fact A_1 and fuzzy rule R , while $(R \otimes A_2)^{1(x_1)}$ computes the generalized modus tollens.

Combination being associative, commutative, and idempotent, we can compute an equivalent problem that is arc consistent using a local propagation scheme. First consider each pair (R_i, x_j) and update A_j if A_j has been modified, propagate the modification on all the variables depending on x_j , i.e., consider each pair (R_k, x_k) such that R_k relates x_k to x_j . If A_k has been modified, propagate the modification similarly. The process is repeated until no modification occurs. This procedure has been implemented by means of an algorithm (F-AC3) that is an extended fuzzy version of the classical AC3 algorithm [26], which is itself more efficient than the original Waltz' [38] algorithm. Note that the classical AC3 algorithm handles only binary relations. Here F-AC3 deals with relations of any arity. It improves the arc consistency of fuzzy CSP with fuzzy relations involving any number of variables.

$$\text{F-AC3 } \{R_1, \dots, R_m\}$$

- 1) $Q := \{(R_i, x_j) \mid R_i \in \{R_1, \dots, R_m\} \text{ and } x_j \text{ restricted by } R_i\}$
- 2) While $Q \neq \emptyset$, do
 - 2.1 Choose and remove a pair (R_i, x_j) from Q
 - 2.2 Compute the restriction induced on x_j via R_i : $A'_j = \text{Revise}(R_i, x_j)$

2.3 if $\text{height}(A_i) = 0$, stop/* Cons(P) = 0: a contradiction is detected */

Else

if $A_i \neq A'_i$,

$$\text{update } A_j: A_j := A'_j;$$

propagate the new restriction on x_j to all the neighbors:

For all R_k relating x_j to other variables,

for all x_i restricted by R_k , $x_i \neq x_j$, add (R_{ik}, x_j) to \mathcal{Q}

$$\text{Revise}(R_i, x_j)$$

$$(R_i \otimes (\bigotimes_{x \text{ s.t. } x \in V(R_i)} A_i)) \downarrow \{x_j\}$$

Since each variable x_i ranges on a discrete set of values U_i , this algorithm stops in a finite number of steps: its worst-case complexity is $O(pmd^{r+1}m)$, m being the number of constraints, p the number of distinct satisfaction degrees effectively used to describe the preferences, r the maximal arity of the constraints, and d the maximum cardinality of sets of possible values U_i . Indeed, the cost of procedure $is_Q(d)$. Initially, Q contains $m \cdot r$ pairs. New entries are made in Q when a call to $revise$ has succeeded. In classical CSP, $revise(R_i, x_i)$ is called with success at most d times. In an FCSP, $revise(R_i, x_i)$ is called with success at most $p \cdot d$ times: since each possible value for x_i may have its degree diminished at most p times: combination is idempotent, decreasing (when combining more and more relations) and does not generate other levels than the original ones. Successful calls of “ $revise$ ” concern at least one of the d possible values for x_i . Hence, theoretical complexity of FAC3 is $O(pmd^{r+1}m)$.

The result does not depend on the order in which constraints are examined. However, the choice of this order may enhance the efficiency. For instance, if the hypergraph is a hypertree, we can take advantage of this structure and propagate first up from the leaves to the root and then backward from the root down to the leaves, as proposed in Shafer and Shenoy [36]. In hypergraphs, some heuristic orders of propagation may be used, depending either on the domain or on the formal characteristic of the problem: propagate first the more restrictive unary relations, propagate first through the tightest constraints, etc.

In fact, our hypergraph representation is similar of the one proposed by Shafer and Shenoy [36] or Dubois and Prade [12] and differs from Pearl's [30] or Fonnck and Straszeka's [18] in the sense that edges are not supposed to be directed and may represent dependencies other than causal ones. Contrary to algorithms proposed by Pearl [30], Shafer and Shenoy [36], Fonnck and Straszeka [18], or Kruse and Schwecke [23] our propagation scheme is not limited to hypertrees. However, it just computes, for each variable, an upper approximation of the marginalization of ρ to each variable. Moreover, the approximation is exact if the hypergraph is in fact a hypertree (proof is similar of the one given by Kruse and Schwecke [23] since our propagation scheme is an extension of theirs from hypertree to hypergraphs).

Contrary to these algorithms, F-AC3 does not control the propagation of the fuzzy unary restrictions. As a consequence, a piece of information A_j or R_j may be combined with itself, for instance after being propagated via a circuit. Hence, the property of idempotence plays a great role in ensuring that propagation does

not generate meaningless information. Moreover, the result of conjunctive combination can only decrease in the sense of fuzzy set inclusion and no new level of satisfaction can appear. These properties ensure the termination of the algorithm. In fact, other formalisms that use combination operations that do not verify these important properties (for instance, Dempster rule of combination) cannot be addressed as simply by local propagation algorithms in structures other than hypertrees.

7.4.2 Making Induced Restrictions Explicit by Ensuring Path Consistency

Path consistency addresses the problem of computing the restrictions induced over pairs of variables by the global problem. This kind of computation may be useful, e.g., in problems involving temporal constraints [11].

For instance, consider the problem $P = \{R_{1,2}, R_{2,3}, R_{1,3}\}$, restricting the possible values of variables $\{x_1, x_2, x_3\}$:

$$R_{12}: x_1 \leq x_2 \quad R_{23}: x_2 \leq x_3 \quad R_{13}: x_1 \geq x_3$$

Constraints R_{12} and R_{23} induce a restriction on x_1 and x_3 : $x_1 \leq x_3$; therefore, the constraint between x_1 and x_3 becomes " $x_1 \leq x_3$ and $x_1 \geq x_3$ ", i.e., R'_{13} : " $x_1 = x_3$." The concept of path consistency introduced in classical CSP.

The concept of path consistency introduced in classical CSP literature by Montanari [29], captures this notion of transitivity in constraint graphs where restrictions are unary or binary. Let us denote R_{ij} the fuzzy restriction linking x_i and x_j , and $R_{ii} = A_i$ the unary restriction on x_i . In terms of fuzzy constraint networks, the original definition can be extended as follows: a problem P is "path consistent" if and only if every pair of values (u_i, u_j) allowed by a direct constraint R_{ij} for the pair of variables (x_i, x_j) with the satisfaction degree $\mu_{R_{ij}}(u_i, u_j)$, is also allowed with a degree at least equal by every path from x_i to x_j . That is, all the restrictions induced by these paths are subsumed by the direct constraints.

Moreover, the graph can always be considered as complete, adding noninformative restrictions between pairs of variables that are not linked. Under this assumption, the consistency of each direct constraint with every path of length 2 implies path consistency. This condition, which is one of transitive closure, is formally expressed by

$$A_i \otimes R_{ij} \otimes A_j \subseteq [R_{ik} \otimes A_k \otimes R_{kj}]^{1(x_i, x_j)} \quad (2)$$

This result is due to Mackworth [26] when relations are crisp and may be easily extended to fuzzy binary relations (by straightforward induction on the length of the path). Hence, a procedure that considers the graph as complete and transforms the problem into an equivalent one that verifies property (2) ensures path consistency. Such a procedure is similar to a shortest path procedure.

Going back to our example, $R_{1,3}$ has been replaced by $R_{1,3}$. This new restriction can be propagated: the restriction $x_1 \geq x_2$ is induced from $R_{1,3}$ and $R_{2,3}$. Intersecting with $R_{1,2}$, the restriction $R'_{1,2}: x_1 = x_2$. Repeating the process, we obtain $R_{3,3}: x_2 = x_3$. That is, P has been transformed into an equivalent problem $P' = \{R'_{1,2}, R_{3,3}, R'_{1,3}\}$ in which all the constraints are explicit.

The following algorithm which is a generalization to fuzzy restrictions of PC2 [26] ensures both arc and path consistency in graphs of binary fuzzy restrictions. Since it takes more induced restrictions into account than a simple arc consistency algorithm, each unary constraint A_j is a better approximation of $\rho^{\dagger}(x_j)$.

F-PC2

\langle number all the variables from 1 to $n \rangle$
 $Q \leftarrow \{(i, k, j) / i \leq j, \text{not}(i = j = k)\}$
 While $Q \neq \emptyset$ do:

 $\langle \text{choose and remove } (i, k, j) \text{ in } Q \rangle$

Modified := false

$$R'_{ij} := R_{ij} \otimes [R_{ik} \otimes A_k \otimes R_{kj}]^{\downarrow (x_i, x_j)}$$

If $\text{height}(R'_{ij}) = 0$, stop/* Cons(P) = 0: a contradiction is detected*/

else if $R'_{ij} \neq R_{ij}$, do

$$R_{ij} := R'_{ij}$$

Modified := true

If Modified, then $Q \leftarrow Q \cup \text{related-paths}(i, k, j)$.

Related-paths (i, k, j):

$$\text{if } i < j, \{(i, j, m)/i \leq m, m \neq j\} \cup \{(m, i, j)/m \leq j, m \neq i\} \cup$$
$$\{(j, i, m) / j < m\} \cup \{(m, j, i) / m < i\}$$

else if $i = j$, $\{(p, i, m)/p \leq m, \text{not}(p = i = m), \text{not}(p = m = k)\}$

As for F-AC3 some heuristics on the ordering of nodes may be used for improving F-PC2. The complexity of the classical PC2 algorithm, which takes advantage of the finite character of variable domains is $O(d^5m^3)$ [26]. The complexity of F-PC2 is thus bounded by $O(d^3m^3h)$, since, like F-AC3, the F-PC2 algorithm is directly inspired from the corresponding classical CSP algorithm.

Note that the updating pattern $R_{ij} := R_{ij} \otimes [R_{i_k} \otimes A_k \otimes R_{k_j}]^{\perp_{(x_i, x_j)}}$ encodes i modus ponens, i modus tollens, and a form of resolution principle (i.e., involving only three variables). Indeed, if $i = j$, the updating pattern becomes $A_i := A_i \otimes [R_{i_k} \otimes A_k]^{\perp_{(x_i)}}$, which is nothing but the inference pattern of arc consistency that encodes both modus ponens and modus tollens. On the other hand, suppose that p , q , and r are three boolean variables restricted by the following constraints:

AP: $p \in \{t, f\}$; AQ: $q \in \{t, f\}$; AR: $r \in \{t, f\}$ are crisp constraints

Kpq : " p or $\neg q$ (priority a)" represented by

$$\mu_{K_{pq}}(p, q) = 1 - \alpha \text{ if } p = f \text{ and } q = t, \mu_{K_{pq}}(p, q) = 1 \text{ otherwise,}$$

Rqr : “ q or r (priority b)” represented by

$$\mu_{R_{qf}}(q, r) = 1 - b \text{ if } q = f \text{ and } r = f, \mu_{R_{qf}}(q, r) = 1 \text{ otherwise;}$$

$[R_{i_k} \otimes A_k \otimes \kappa_{i_j}]^{\uparrow(x_k, x_j)}$ is the fuzzy restriction corresponding to the fuzzy set defined by $\mu_{R_{i_k}}(p, r) = \max(1 - a, 1 - b)$ if $p = f$ and $r = f, H_{R_{i_k}}(p, r) = 1$ otherwise. In other words, path consistency computes the resolution pattern in possibilistic logic, namely

$$\frac{(p \vee \neg q a); (q \vee r b)}{(p \vee r \min(a, b))}$$

a, b being lower bounds of necessity degrees [9].

7.5 FINDING THE BEST SOLUTION

The previous algorithms calculate upper approximations of $\rho^{\dagger(x)}$, which can be exact under some structural conditions. If the approximation is exact, the minimal height of all the constraints is equal to the consistency of the problem: $\text{Cons}(P) = \min_j \text{height}(R_j)$ where the R_j are the relations explicated after applying arc and/or path consistency, and defining a problem equivalent to P . Otherwise, it is only an upperbound of $\text{Cons}(P)$ and computing the problem consistency requires the determination of a best solution, i.e., an instantiation (u_1, \dots, u_n) such as $\mu_\rho(u_1, \dots, u_n) = \text{height}(\rho)$. Indeed, variables are generally interactive. In fact, the problem of computing $\text{Cons}(P)$ reduces to a sup/min optimization problem of the form (in the formula below, some R_i may be unary):

$$\sup_{(u_1, \dots, u_n) \in \mathcal{U}_1 \times \dots \times \mathcal{U}_n} (\min_{R_i \in \{R_1, \dots, R_m\}} (\mu_{R_i}((u_1, \dots, u_n)^{\downarrow V(R_i)})))$$

Following Lang [24] or Schex [35], such a problem may be solved in the finite case using classical tree-search algorithms, namely Depth first, Branch and Bound, α - β , SSS*, beam search, etc.

Using a classical tree search algorithm, variables are instantiated in a prescribed order $\langle x_1, \dots, x_n \rangle$. The root of the tree is the empty assignment. Intermediary nodes denote partial instantiations. Leaves represent complete instantiations of $\langle x_1, \dots, x_n \rangle$, i.e., potential solutions. In a depth-first exploration of the tree, we will keep track of the current leaves that maximize U_θ .

The use of flexible constraints makes it possible to prune each branch that will necessarily lead to complete instantiations whose degree of satisfaction is worse than that of the best of the already evaluated solutions. In other terms, it is useless to extend intermediary nodes u_1, \dots, u_k such that $\mu_{[u_1, \dots, u_k]}(u_1, \dots, u_k) < \alpha$, α being the current lower bound of $\text{Cons}(P)$. Threshold α may be initialized by 0 and updated when a solution (u_1, \dots, u_n) such as $\mu_\rho(u_1, \dots, u_n) > \alpha$ is reached.

The computation of $\mathcal{H}_{\{u_1, \dots, u_k\}}(u_1, \dots, u_k)$ is too costly [since it requires the computation of all the extensions of (u_1, \dots, u_k) to (x_1, \dots, x_n)]. So in practice, we compute an upper bound of $\mathcal{H}_{\{u_1, \dots, u_k\}}(u_1, \dots, u_k)$ instead. We call this upper bound the “consistency” of the partial instantiation (u_1, \dots, u_k) . The consistency of (u_1, \dots, u_k) considers only the restrictions pertaining to the variables instantiated

in (u_1, \dots, u_k) , i.e., constraints R_i such that $V(R_i) \subseteq \{x_1, \dots, x_k\}$:

$$\text{Cons}(u_1, \dots, u_k) = \min_{R_i \in \mathcal{R}, V(R_i) \subseteq \{x_1, \dots, x_k\}} (\mu_{R_i}(u_1, \dots, u_k))^{1/V(R_i)}$$

This bound decreases when extending the nodes and becomes exact for complete instantiations. Moreover, it may be incrementally computed as the tree is explored downward:

$$\text{Cons}(u_1, \dots, u_k, u_{k+1}) = \min(\text{Cons}(u_1, \dots, u_k), \min_{\{R_i/R_{k+1} \in V(R_i) \text{ and } V(R_i) \subseteq \{x_1, \dots, x_{k+1}\}\}} (\mu_{R_i}(u_1, \dots, u_{k+1}))^{1/V(R_i)})$$

It is interesting to notice that the incremental computation of $\text{Cons}(u_1, \dots, u_n)$ considers each constraint only once.

Let (u_1, \dots, u_k) denote the current partial instantiation, α the current lower bound of $\text{Cons}(P)$, and β an upper bound of the satisfaction degree of the solution searched for [generally β is an upper bound of $\text{cons}(P)$, $\beta = 1$ by default]. Node (u_1, \dots, u_k) is extended as follows for all u_{k+1} : $\text{Cons}(u_1, \dots, u_k, u_{k+1})$ is computed. Only the nodes $(u_1, \dots, u_k, u_{k+1})$ such that $\text{Cons}(u_1, \dots, u_k, u_{k+1}) \geq \alpha$ are created, and the next to be developed is the one such that $\text{Cons}(u_1, \dots, u_k)$ is maximal.

Extend $((u_1, \dots, u_k), \alpha, \beta)$

If $k = n$,/* a solution with satisfaction degree β has been reached*/
best-current-solution := $((u_1, \dots, u_k), \beta)$

return β

Else,/* extend current instantiation*/

for all u_{k+1} in U_{k+1} do

$\beta' := \beta$

for all R_i such as $V(R_i) \subseteq \{x_1, \dots, x_{k+1}\}$ and $x_{k+1} \in V(R_i)$ do

$$\beta' := \min(\beta', \mu_{R_i}(u_1, \dots, u_{k+1})^{1/V(R_i)})$$

if $\beta' > \alpha$, do/

$$\beta := (\min \beta', \text{extend}((u_1, \dots, u_{k+1}), \alpha, \beta))$$

$$\alpha := \max(\alpha, \beta')$$

if $\alpha = \beta$ return β /* α may not be $> \beta$: one of the best sons has been extended */
return α

The search stops when no more node can be created. It is successful if a complete instance has been reached, and the best among those that have been reached is optimal [$\alpha = \text{Cons}(P)$]. If no complete instance has been reached, $\text{Cons}(P) = 0$ and the problem is unfeasible.

Moreover, the search stops as soon as a solution of a degree greater than or equal to β is obtained. Therefore, if a satisfaction level b is considered as sufficient for the solution you fix $\beta = b$. Notice that the best the bounds α and β , the more efficient the search. Moreover, some heuristics may be applied to this tree search, namely on the order of instantiation (instantiate first the more restricted variables, e.g., the variables connected to many others or those having a fuzzy set of possible values with minimal cardinality, i.e., giving preference to the most restricted variables). In fact, it is possible to develop a large class of tree search algorithms (e.g., beam search) based on the same principle and integrating different heuristics or variants.

7.6 EXAMPLE

A tutorial is supposed to be made of lectures on topics P and Q , plus training sessions and computer sessions for topic P . Each type of activity must not involve more than 5 sessions. There should be approximately 8 lectures overall, and the person in charge of lectures and computer sessions on topic P proposes about 6 sessions overall. There should be about two more computer sessions than training sessions, and exactly as many lectures on P as training sessions. The FCSP is defined in Figure 7.3. A 5-level satisfaction scale is used $L = \{0, a, b, c, 1\}$. Representing fuzzy sets by pairs $(u, \mu_P(u))$, we have the following representations of "about 8" (E), "about 6" (B), and "about 2" (C): $E = \{(6, a), (7, c), (8, 1), (9, c), (10, a)\}$; $B = \{(5, b), (6, 1), (7, b)\}$; $C = \{(1, b), (2, 1), (3, b)\}$; $I = \{0, 1, 2, 3, 4, 5\}$. With the following understanding of the variables, x (resp. y): number of lectures on P (resp. Q); w : number of computer sessions on P ; z : number of exercise sessions, the constraints write $Ax: x \in I$; $A_1y: y \in I$; $Az: z \in I$; $A_1w: w \in I$; $Rxy: x + y \in E$; $Rxw: x + w \in B$; $Rwz: w - z \in C$; $Rxz: x = z$. [Here the sign \in means that the (fuzzy) subset, which is after the sign, restricts the possible values of the variable written before the sign.]

Arc consistency performed on x via Rxy leads to $\mu_{Rxy-x}(u) = \max_{y=1,5} \mu_E(u+y)$, $\forall u$, which gives $I \cap Rxy \rightarrow x = \{(1, a), (2, c), (3, 1), (4, 1), (5, 1)\} = A_x$. Clearly more precision has been obtained on x since only 3, 4, 5 remain totally possible values: for instance the satisfaction degree of value 2 have been reduced

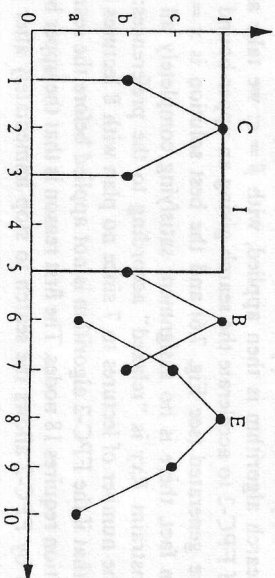


FIGURE 7.3. The constraints in the example.

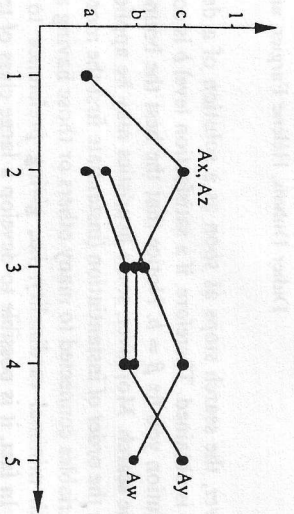


FIGURE 7.4. Results using arc consistency only.

since the best assignment of y ($y = 5$) corresponding to $x = 2$ satisfies R_{xy} only at degree c .

Note that the computation of $R_{xy} \rightarrow x$ could be done via fuzzy arithmetics [10]: the restriction $R_{xy} \rightarrow x = (R_y \otimes R_{xy})^{1(x)}$ induced on x is in fact: $x \in E \ominus I$, \ominus denoting the subtraction of fuzzy quantities.

Applying FAC-3 to this example (see Fig. 7.4), we discover that the height of each unary restriction is c , i.e., that constraints are conflicting [c is an upper approximation of $\text{Cons}(P)$]. Note that the unary restrictions are only upper approximations of the marginals of the global relation. For instance, $x = 4$ is allowed with degree b , although there is in fact no assignment of x, y, z, w with $x = 4$ satisfying all the constraints even partially.

Consider now arc + path consistency instead of arc consistency only. R_{xz} states that $x = z$. The path x, w, z states that $R_{xw}: x + w \in B$ and $R_{wz}: w - z \in C$. Combining these restrictions, we obtain the induced restriction $(R_{xw} \otimes R_{wz} \otimes A_w)^{1(x,z)}$. Because of A_w , this restriction is more restrictive than $x + z \in B \ominus C = (R_{xw} \otimes R_{wz})^{1(x,z)}$. In this example, path consistency performs nothing but variable elimination. Intersecting this restriction then with the set R_{xz} of possible pairs of values for x and z , we get for R'_{xz} : $\mu_{R'_{xz}}(2, 2) = 1$, $\mu_{R'_{xz}}(1, 1) = \mu_{R_{xz}}(3, 3) = b$, $\mu_{R'_{xz}}(x, z) = 0$ otherwise. The new restriction is then propagated on x by arc consistency: $\mu_{R_{xz}}(2) = c$, $\mu_{R_{xz}}(x) = 0$ otherwise (note that value $x = 1$ is ruled out). Applying F-PC2 instead of FAC-3 to the example (see Fig. 7.5), the restrictions obtained for x, y , and z are tighter. Moreover, they are equal to the marginals of the global relation. Again the minimal height of the constraints (c) provides an upper bound of $\text{Cons}(P)$.

The search algorithm is then applied, with $\beta = c$: we take advantage of the results of FPC-2 to accelerate the search giving an upper-bound of $\text{Cons}(P)$. Ten nodes are generated (see Fig. 7.6) and the best solution is $x = 2, y = 5, z = 2, w = 4$. In fact, there is no assignment satisfying completely all the constraints; thus, constraint R_{xy} is "relaxed" according to the preferences: one accepts to reduce the number of lectures to 7 since no plan with 8 lectures is feasible.

Note that if the FPC-2 algorithm is not applied before the search, reaching the best solution requires 18 nodes. The first reason is that the upper bound of $\text{Cons}(P)$ provided by FPC-2 allows the search to stop immediately after the discovery of

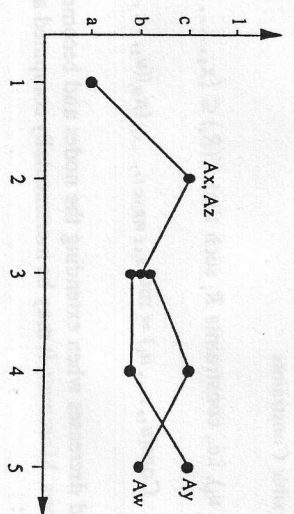


FIGURE 7.5. Results using arc and path consistency.

the solution. Moreover, the problem obtained by FPC-2 contains more explicit constraints than the original one; thus, the consistency of a partial instantiation is a better upper bound of the satisfaction level of its best extension: more useless branches are pruned.

The previous example involves imperative constraints only. Let us study a case in which a constraint has less priority than the others: suppose that the constraint relating x and z ("exactly as many lectures on P as training sessions") has a priority $d < a = 1 - c$; the fuzzy relation modeling this constraint is

$$R_{xz}: \mu_{R_{xz}}(x, z) = 1 \text{ if } x = z; \quad \mu_{R_{xz}}(x, z) = 1 - d \text{ otherwise}$$

The FCSP is still partially inconsistent but may be solved by relaxing R_{xz} : the consistency of the problem is $1 - d$ and the best solution is an instantiation satisfying all the constraints but R_{xz} ($x = 3, y = 5, z = 1, w = 3$). Note that the weakening of the priority of R_{xz} , i.e., the constraint $x = z$ leads to a solution with x quite different from z ; this exemplifies the distinction between prioritized and soft constraints; only soft constraints guarantee a progressive degradation of the global satisfaction.

7.7 NONMONOTONICITY IN FCSPs

Using the classical CSP approach, the set of solutions diminishes when new constraints are added, and eventually becomes empty in case of conflicting

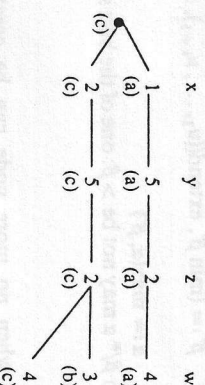


FIGURE 7.6. Search tree in the example.

constraints. In the FCSP framework, adding a new constraint to a problem P may rule out the previous best solutions if they satisfy the new constraint to a degree lower than $\text{Cons}(P)$. But as long as the new problem (say P') is not totally inconsistent, a new set of best solutions appears that satisfies the new problem to a degree $\text{Cons}(P') \leq \text{Cons}(P)$. Indeed, we have

$$R_1 \otimes \cdots \otimes R_m \otimes R_{m+1} \subseteq R_1 \otimes \cdots \otimes R_m$$

where \subseteq stands for the fuzzy set inclusion, but generally

$$\begin{aligned} \{u_1, \dots, u_n\} / \mu_{R_1} \otimes \cdots \otimes \mu_{R_m} \otimes \mu_{R_{m+1}}(u_1, \dots, u_n) \\ = \text{Cons}(P') \neq \{u_1, \dots, u_n\} / \mu_{R_1} \otimes \cdots \otimes \mu_{R_m}(u_1, \dots, u_n) = \text{Cons}(P) \end{aligned}$$

Hence, the set of best solutions does not decrease monotonically when new constraints are added. The nonmonotonic behavior of soft constraint has been noticed by Satoh [34]. The type of nonmonotonicity at work here is the same as the one captured by possibilistic logic and appears only in the presence of inconsistency. It has been precisely characterized by Benferhat et al. [2] as the class of preferential inferences satisfying the rational monotony property of Lehmann [25]. In fact, adding a new constraint may lead to four situations:

1. The new constraint is redundant: $R_1 \otimes \cdots \otimes R_m \subseteq R_{m+1}$; the set of best solutions remains unchanged.
2. The new constraint is totally compatible with P : $\text{Cons}(P) = \text{Cons}(P')$; the set of best solutions is included in the previous one but may remain unchanged.
3. The new constraint is partially inconsistent with P : $\text{Cons}(P') < \text{Cons}(P)$; constraints are implicitly relaxed according to their flexibility and the set of best solutions is not generally included in the previous one.
4. The new constraint is totally incompatible with P : $\text{Cons}(P') = 0$; the set of (best) solutions is empty.

As a consequence, the problem of solution maintenance in dynamic CSPs appears to be more complex than in classical CSPs: pruned branches of a previous search in the tree have to be developed contrary to the method proposed by Van Hentenryck [37].

The question of relaxing or deleting a constraint is not separately considered in the FCSP model, since the relaxation capacities of the constraints are supposed to be explicitly represented by means of preferred values and priority degrees. In other terms, constraints can be only added or strengthened, i.e., the priority of a constraint (resp. the satisfaction degree of a value) may dynamically increase (resp. decrease) but not decrease (resp. increase). As seen in the example of Section 7.6, the decrease of the priority of constraint R_{xz} from 1 to $d < 1 - c$ opens the door to a completely different solution, since we then have to deal with a different problem.

7.8 CONCLUSION

Flexible constraints have two advantages over crisp ones: the different solutions are rank-ordered in accordance to their degrees of feasibility and the search algorithm takes advantage of this flexibility to efficiently focus on the best solutions. Moreover, the FCSP approach may handle partial inconsistencies between constraints and provides a solution (the best one) as long as the problem is not totally inconsistent.

Thus the FCSP framework seems to be a suitable approach to handle both soft constraints, i.e., constraints involving preferences as well as prioritized constraints—and more generally problems involving fuzzy relations. It marries possibility theory and classical CSP theory. Possibility theory offers its rich and powerful setting for the representation of soft and prioritized constraints; moreover, the classical operations of combination and projection of fuzzy restrictions present a low computational cost, and properties (especially idempotence) allow the use of local computational schemes on general structures. Moreover, the CSP literature provides theoretical results as well as efficient algorithms of constraint propagation, which can be easily adapted to the treatment of fuzzy restrictions. Finally, this formalism suggests a nonmonotonic framework for dynamic problems, for instance, the use of default constraints.

In terms of applications, FCSP seems particularly promising in areas such as scheduling, planning, and design, which often make use of classical CSP techniques mixed with informal approaches to flexibility (see for instance Descottes and Latombe [7]). Note that the framework we have described handles only variables ranging on discrete domains; in order to handle some of these applications, it must be extended to continuous domains (see Davis [5]).

On the other hand, the FCSP approach may be regarded as a problem-solving paradigm for every application involving a set of fuzzy restrictions, especially problems of combination of vague information, fuzzy relational databases, or qualitative reasoning (e.g., [13]). In particular, it should be noticed that flexible constraints are coherent with possibilistic logic: FCSP may be a semantic approach to possibilistic logic. Indeed, a necessity-valued formula may be regarded as an elastic constraint on the possible boolean values that can be assigned to boolean variables [15].

References

1. R. E. Bellman and L. A. Zadeh. Decision-making in a fuzzy environment. *Manage. Sci.* 17, 141–164, 1970.
2. S. Benferhat, D. Dubois, and H. Prade. Representing default rules in possibilistic logic. *Proc. 3rd Int. Conf. Principles of Knowledge Representation and Reasoning* (KR'92). Cambridge, Mass., 673–684, 1992.
3. A. Borning, M. Maher, A. Marindale, and M. Wilson. Constraint hierarchies and logic programming. *Proc. Int. Conf. Logic Programming*, Lisbon, Portugal, 149–164, 1989.
4. J. Bowen, R. Lai, and D. Bahner. Fuzzy semantics and fuzzy constraint networks. *Proc. 1st IEEE Int. Conf. Fuzzy Syst.*, San Diego 1009–1016, 1992.

5. E. Davis. Constraint propagation with interval labels. *Artificial Intelligence* 32, 281-331, 1987.
6. R. Dechter and J. Pearl. Tree clustering for constraint networks. *Artificial Intelligence* 38, 353-366, 1989.
7. Y. Descottes and J. C. Latombe. Making compromises among antagonist constraints. *Artificial Intelligence* 27, 149-164, 1985.
8. D. Dubois. An application of fuzzy arithmetic to the optimization of industrial machining processes. *Math. Model.* 9(6), 461-475, 1987.
9. D. Dubois, J. Lang, and H. Prade. Automated reasoning using possibilistic logic: Semantics, belief revision and variable certainty weights. *Proc. 5th Workshop on Uncertainty in Artificial Intelligence*, Windsor, Aug. 18-20, 81-87, 1989. Revised version to appear in *IEEE Trans. Data Knowledge Engin.* 6(1), 1994.
10. D. Dubois and H. Prade (with the collaboration of H. Farreny, R. Martin-Clouaire, and C. Testemale. *Possibility Theory—An Approach to Computerized Processing of Uncertainty*. New York: Plenum Press, 1988.
11. D. Dubois and H. Prade. Processing fuzzy temporal knowledge. *IEEE Trans. Syst. Man Cybernet.* 19(4), 729-744, 1989.
12. D. Dubois and H. Prade. Inference in possibilistic hypergraphs. In *Uncertainty in Knowledge Bases*, B. Bouchon-Meunier, R. R. Yager, and L. A. Zadeh, eds. Lecture Notes in Computer Science, Vol. 521. Berlin: Springer-Verlag, 1991, 250-259.
13. D. Dubois and H. Prade. Semantic considerations on order of magnitude reasoning. In *Decision Support Systems and Qualitative Reasoning*, M. G. Singh and L. Travé-Massuyès, eds. Amsterdam: North-Holland, 1991, 223-228.
14. D. Dubois and H. Prade. Possibilistic logic, preferential models, non-monotonicity and related issues. *Proc. 12th Int. Joint Conf. Artificial Intelligence (IJCAI'91)*, Sydney, Australia, Aug. 24-30, 419-424, 1991.
15. D. Dubois and H. Prade. Fuzzy rules in knowledge-based systems—Modelling gradedness, uncertainty and preference. In *An Introduction to Fuzzy Logic Applications in Intelligent Systems*, R. R. Yager and L. A. Zadeh, eds. Boston: Kluwer Academic Publ., 1992, 45-68.
16. B. Faltings, D. Haroud, and I. Smith. Dynamic constraint propagation with continuous variables. *Proc. 10th Eur. Conf. Artificial Intelligence*, Vienna, Aug. 3-7, 754-758, 1992.
17. B. N. Freeman-Benson, J. Maloney, and N. Borning. An incremental constraint solver. *Commun. ACM* 33(1), 54-63, 1990.
18. P. Fonck and E. Straszcka. Building influence networks in the framework of possibility theory. *Ann. Univ. Sci. Budapest. Sect. Comp.* 12, 101-106, 1991.
19. E. C. Freuder. Partial constraint satisfaction. *Proc. 11th Int. Joint Conf. Artificial Intelligence*, Detroit, Michigan, Aug. 20-25, 278-283, 1989.
20. E. C. Freuder, R. J. Wallace. Partial constraint satisfaction. *Artificial Intelligence*, 58, 21-70, 1992.
21. J. Kacprzyk and S. A. Orlovski. Fuzzy optimization and mathematical programming: A brief introduction and survey. In *Optimization Models Using Fuzzy Sets and Possibility Theory*, J. Kacprzyk and S. A. Orlovski, eds. Dordrecht: D. Reidel, 1987, 50-72.
22. R. L. Keeney, and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. New York: John Wiley, 1976.
23. R. Kruse and E. Schwecke. Fuzzy reasoning in a multidimensional space of hypotheses. *Int. J. Approximate Reasoning* 4, 47-68, 1990.
24. J. Lang. Possibilistic logic as a logical framework for min-max discrete optimization problems and prioritized constraints. In *Fundamentals of Artificial Intelligence Research* (Int. Workshop, Smolenice, Czechoslovakia, Sept. 8-12, 1991). Ph. Jorrand and J. Kelenen, eds. Lecture Notes in Artificial Intelligence, Vol. 535. Berlin: Springer-Verlag, 1991, 112-126.
25. D. Lehmann. What does a conditional knowledge base entail. *Proc. 1st Int. Conf. Principles Knowledge Representation and Reasoning (KR'89)*, Toronto, 212-221, 1989.
26. A. K. Mackworth. Consistency in networks of relations. *Artificial Intelligence* 8, 99-118, 1977.
27. R. Martin-Clouaire. Dealing with soft constraints in a constraint satisfaction problem. *Proc. 4th Int. Conf. Information Process. Uncertainty Knowledge-Based Sys.* Palma de Mallorca, July 6-10, 37-40, 1992.
28. R. Mohr and T. Henderson. Arc and path consistency revisited. *Artificial Intelligence* 28, 225-233, 1986.
29. U. Montanari. Network of constraints: Fundamental properties and application to picture processing. *Inform. Sci.* 7, 95-132, 1974.
30. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.
31. H. Prade. Reasoning with fuzzy default values. *Proc. 5th Int. Symp. Multiple-Valued Logic*, Kingston, Ontario, May 28-30, 191-197, 1985.
32. G. Qi and G. Friedrich. Extending constraint satisfaction problem solving in structural design. In *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, F. Belli and F. J. Radermacher, eds. Lecture Notes in Artificial Intelligence, Vol. 604. Berlin: Springer-Verlag, 1992, 341-350.
33. N. Sadeh. Look-ahead techniques for micro-opportunistic job shop scheduling. Report CS-91-102, Carnegie Mellon University, 1991.
34. K. Satoh. Formalizing soft constraint by interpretation ordering. *Proc. 9th Eur. Conf. Artificial Intelligence (ECAI'90)*, Stockholm, 585-590, 1990.
35. T. Schex. Possibilistic constraint satisfaction problems or how to handle soft constraints. *Proc. 8th Conf. Uncertainty Artificial Intelligence*, Stanford, July 17-19, 268-275, 1992.
36. G. Shaler and P. Shenoy. Local computation in hypertrees. Working Paper No. 201, School of Business, Kansas University, 1988.
37. P. Van Hentenryck. Incremental constraint satisfaction in logic programming. *Proc. Int. Conf. Logic Programming (ICLP'90)*, 189-202, 1990.
38. D. Waltz. Understanding line drawings of scenes with shadows. In *The Psychology of Computer Vision*, P. Winston, ed. New York: MacGraw-Hill, 19-91, 1975.
39. K. L. Wood and E. K. Antonsson. Computations with imprecise parameters in engineering design: Background and theory. *ASME J. Mech. Transmiss. Automation Design* 111(4), 616-625, 1989.
40. L. A. Zadeh. Calculus of fuzzy restrictions. In *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*, L. A. Zadeh, K. S. Fu, K. Tanaka, and M. Shimura, eds. New York: Academic Press, 1975, 1-39.
41. L. A. Zadeh. The concept of linguistic variable and its application to approximate reasoning. *Inform. Sci.* Part 1: 8, 199-249; Part 2: 8, 301-357; Part 3: 9, 43-80, 1975.
42. L. A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets Syst.* 1, 3-28, 1978.
43. H. J. Zimmermann. Fuzzy programming and linear programming with several objective functions. *Fuzzy Sets Syst.* 1, 46-55, 1978.