Constraints and preferences
The interplay of preferences and algorithms

T. Schiex
INRA
Toulouse, France)

M. Cooper
IRIT
Toulouse, France

Abstract

As logic, constraint satisfaction faces the problem of inconsistency which itself naturally leads to the need of expressing preferences. Starting from (Rosenfeld, Hummel, & Zucker 1976), which defined fuzzy constraint networks, a variety of extended constraint frameworks have been proposed. Since 1995, several general axiomatic frameworks that try to cover all these proposals have been introduced.

In this paper, we show how algorithms and axioms interacts on a specific class of algorithms: arc consistency enforcing algorithms. The generalization of arc consistency to the *Valued CSP* framework was recently made possible thanks to the addition of an additional axiom that captures the existence of difference between preferences. We show that many usual (and less usual) instances satisfy this axiom. This new axiom naturally suggests a modification of the set of axioms that could simplify both the axiom sets, the algorithms and the proofs on Valued CSP. It consists in shifting from a semi-group to a full group, where the existence of an opposite is guaranteed. We consider this alternate definition and show that it leads to a strong reduction of the framework generality.

The constraint satisfaction framework faces, as logic, the problem of inconsistency. It is therefore not surprising that the notion of preferences has been introduced very early in the history of constraint satisfaction, going back to (Rosenfeld, Hummel, & Zucker 1976). Most, if not all, existing frameworks are based on the idea that a mathematical object (a preference) can be generated by each constraint when variables are assigned. When several constraints are assigned, preferences must be *combined*. In order to maximize preferences, one must also be able to *compare* combined preferences.

There are many types of preferences that captures different practical problems and that corresponds to distinct properties. To try to capture a large variety of such preference schemes, general abstract preferences frameworks have been developed in the constraint satisfaction community (Freuder 1989; Schiex, Fargier, & Verfaillie 1995; Bistarelli, Montanari, & Rossi 1995).

The aims in designing such frameworks are quite different and result from different compromises between :

- generality: expressive power of the abstract framework, what variety of preferences schemes can be captured…
- specificity: what generic properties, theorems and algorithms can be built for handling these preferences.

Indeed, a very interesting situation arises from the interaction of the properties of the preference scheme used and the algorithms that may be used to tackle problems (satisfaction, inference…) using this scheme. Historically, the first such abstract framework, known as “Partial CSP” (Freuder 1989) put the emphasis on generality and essentially ignores this intricate interplay.

When algorithms come in the arena, one obvious aim is to maximize the generality of the preference abstract framework under the constraint that a given class of algorithms works. One class of algorithms that covers a wide spectrum of problems originating from graph theory, operation research and matrix factorization is the class of so-called dynamic programming algorithms (Bertelé & Brioshi 1972), among which shortest path algorithms are probably the most famous. In this case, one seeks to *minimize* a *sum*. For such algorithms, it is now well known (Aho, Hopcroft, & Ullman 1974; Minoux 1976; Shenoy 1991) that a semi-ring structure can be used. The semiring CSP framework uses a large subclass of semirings where one operator is used for “maximization” of preferences and another is used for combination. They allow to express a wide class of preferences among which ones that define partially ordered structures. Naturally, semiring CSP can be perfectly solved by dynamic programming (also known as bucket elimination (Dechter 1999)). However, the most famous class of constraint networks algorithms, the so-called “local consistency enforcing” algorithms, which are extremely useful to solve constraint networks without preferences were shown to terminate only on a very small subclass of semiring CSP: those with an idempotent preference aggregation operator.

At the very same time,in order to avoid redundant developments, to make it possible to use a variety of preference schemes and to better understand this intricate interaction between preferences and algorithms, we developed an algebraic framework for preferences in constraint networks called the *Valued CSP* framework, which is based on a monotonic semi-group structure.

In this paper, we rapidly present the framework, most of its know properties and some classical and less usual preference schemes it may capture. As an example of the intricate interplay between preference structures and algorithms, we recently shown that the addition of a small axiom to the basic VCSP axioms makes it possible to extend the notion of arc consistency to valued CSP. This axiom simply enforces the existence of a (maximal) difference between preferences. The notion of difference is a specialization of existence of the opposite that often leads to intricate proofs and complex conditions. A natural idea is to consider a simple stronger axiom which is the existence of an opposite valuation. We therefore consider the shift from a semi-group to a group structure and show that it leads to a strong loss of generality.

Constraint Networks

For those unfamiliar with it, we rapidly describe what a constraint network is and some usual notions on it.

A constraint satisfaction problem (CSP) is a triple $\langle X, D, C\rangle$.

- X is a set of n variables $X = \{1, \ldots, n\}$.
- Each variable $i \in X$ has a domain of values $d_i \in D$ and can be assigned any value $a \in d_i$, also noted (i, a). d will denote the cardinality of the largest domain of a CSP.
- C is a set of constraints. Each constraint $c_P \in C$ is defined over a set of variables $P \subseteq X$ (called the scope of the constraint) by a subset of the Cartesian product $\prod_{i \in P} d_i$ which defines all consistent tuples of values. The cardinality $|P|$ is the arity of the constraint c_P. r will denote the largest arity of a CSP.

We assume, without loss of generality, that at most one constraint is defined over a given set of variables. The set C is partitioned into three sets $C = C^1 \cup C^+$ where C^1 contains all unary constraints. For simplification, the unary constraint on variable i will be denoted c_i, binary constraints being denoted c_{ij}. $e = |C^+|$ will denote the number of non unary constraints in a CSP. If $J \subseteq X$ is a set of variables, then $\ell(J)$ denotes the set of all possible labellings for J *i.e.*, the Cartesian product $\prod_{i \in J} d_i$ of the domains of the variables in J. The projection of a tuple of values t onto a set of variables $V \subseteq X$ is denoted by $t_{\downarrow V}$. A tuple of values t satisfies a constraint c_P if $t_{\downarrow P} \in c_P$. Finally, a tuple of values over X is a solution iff it satisfies all the constraints in C.

The usual problem on a constraint network is the famous "satisfaction" problem where the aim is to choose a value in the domain of each variable in such a way that, for any constraint c, the combination of values of the variable in the scope of c appear in the set of tuples in the constraint. This is called a solution of the problem.

One the most important notion on classical CSP is the notion of local consistency among which the most important is probably arc consistency:

DEFINITION 1 *Given a binary CSP (X, D, C), a value $a \in d_i$ is said to be viable if $\forall c_{ij} \in C, \exists b \in d_j$ such that $(a, b) \in c_{ij}$. A CSP is arc consistent if all its values are viable.*

It is well know that from any given CSP, one may build an equivalent CSP (with the same set of solutions) which is arc consistent. This so-called arc consistent closure is unique and can be built in time $O(ed^2)$, space $O(ed)$ (Bessière 1991). The process of building such a closure consists in enforcing the deletion of all non viable values until quiescence. This corresponds to the saturation of the constraint network by a local incomplete inference process. Since only logical consequences are produced, the final network is equivalent to the initial one. Arc consistency is at the core of most constraint satisfaction algorithms and systems and its extension to soft constraints frameworks is a natural quest with likely practical and theoretical side-effects.

Valued constraint Networks

Valued CSP (or VCSP) were initially introduced in (Schiex, Fargier, & Verfaillie 1995). A valued CSP is obtained by associating a valuation with each constraint. The set E of all possible valuations is assumed to be totally ordered and its maximum element is used to represent total inconsistency. When a tuple violates a set of constraints, its valuation is computed by combining the valuations of all violated constraints using an aggregation operator, denoted by $\oplus$. This operator must satisfy a set of properties that are captured by a set of axioms defining a so-called *valuation structure*. Valuations actually represent local dislikes (rather than preferences).

DEFINITION 2 *A* valuation structure *is defined as a tuple $\langle E, \oplus, \succcurlyeq\rangle$ such that:*

- *E is a set, whose elements are called valuations, which is totally ordered by $\succcurlyeq$, with a maximum element denoted by $\top$ and a minimum element denoted by $\bot$;*
- *E is closed under a commutative, associative binary operation $\oplus$ that satisfies:*
  - Minimum*: $\forall \alpha \in E, \alpha \oplus \bot = \alpha$;*
  - Monotonicity*: $\forall \alpha, \beta, \gamma \in E, (\alpha \succcurlyeq \beta) \Rightarrow \big((\alpha \oplus \gamma) \succcurlyeq (\beta \oplus \gamma)\big)$;*
  - Maximum*: $\forall \alpha \in E, (\alpha \oplus \top) = \top$.*

When E is restricted to $[0, 1]$, this structure of a totally ordered commutative monoid with a monotonic operator is also known in uncertain reasoning, as a triangular conorm (Dubois & Prade 1982).

DEFINITION 3 *A valued CSP is a tuple $\langle X, D, C, S\rangle$ where X is a set of n variables $X = \{1, \ldots, n\}$, each variable $i \in X$ has a domain of possible values $d_i \in D$. $C = C^1 \cup C^+$ is a set of constraints and $S = \langle E, \oplus, \succcurlyeq\rangle$ is a valuation structure. Each constraint $c_P \in C$ is defined over a set of variables $P \subseteq X$ as a function $c_P : \prod_{i \in P} d_i \rightarrow E$.*

An assignment t of values to some variables $J \subseteq X$ can be simply evaluated by combining, for all assigned constraints c_P (i.e., such that $P \subseteq J$), the valuations of the projection of the tuple t on P:

DEFINITION 4 *In a VCSP $V = \langle X, D, C, S\rangle$, the valuation of an assignment t to a set of variables $J \subseteq X$ is defined by:*

$$\mathcal{V}_V(t) = \bigoplus_{c_P \in C, P \subseteq J} [c(t_{\downarrow P})]$$

Note that, thanks to monotonicity, this is a lower bound on the valuation of any assignment that assigns all variables of the problem. This property is very useful in order to solve the central problem of VCSP: finding a complete assignment with a minimum valuation.

Globally, the semantics of a VCSP is defined by the valuations $\mathcal{V}(t)$ of assignments t to X. The choice of axioms is quite natural and is usual in the field of uncertain reasoning. The ordered set E simply allows us to express different degrees of constraint violation. The commutativity and associativity guarantee that the valuation of an assignment is independent of the order in which valuations are combined. The monotonicity of $\oplus$ guarantees that assignment valuations cannot decrease when constraint violations increase. For a more detailed analysis and justification of the VCSP axioms, we invite the reader to consult (Schiex, Fargier, & Verfaillie 1995) which also emphasize the difference between idempotent and strictly monotonic aggregation operators $\oplus$.

DEFINITION 5 *An operator* $\oplus$ *is idempotent if* $\forall \alpha \in E, (\alpha \oplus \alpha) = \alpha$. *It is strictly monotonic if* $\forall \alpha, \beta, \gamma \in E, (\alpha \succ \beta) \wedge (\gamma \neq \top) \Rightarrow (\alpha \oplus \gamma) \succ (\beta \oplus \gamma)$

As shown in (Schiex, Fargier, & Verfaillie 1995), these two properties are incompatible as soon as $|E| > 2$. The only valuation structures with an idempotent operator correspond to classical and possibilistic CSP (Schiex 1992) (min-max dual to the conjunctive fuzzy CSP framework) which use $\oplus = \max$ as the aggregation operator. Other soft CSP frameworks such as MAX-CSP, lexicographic CSP or probabilistic CSP use a strictly monotonic operator.

Extending Arc consistency

In classical CSP, enforcing arc consistency is a process that allows to transform an initial CSP to an equivalent CSP that satisfies a local consistency property (the very arc consistency property). Arc consistency enforcing satisfy some crucial properties: it preserves the semantics of the problem, it always terminates, in polynomial time and it defines a unique equivalent problem called the arc-consistent closure (the filtering process is confluent).

When abstract frameworks such as VCSP and semi-ring CSP were introduced, it was immediately proved in each case that the arc consistency property and associated enforcing algorithms could be extended to abstract frameworks as far as the operator that combines valuation is idempotent. In this case, all the usual good properties of arc consistency (termination, polynomial time, semantics preservation, confluence) are kept.

However, when the operator is not idempotent, various extensions of arc consistency were defined that either were enforced by non terminating algorithms that did not yield an equivalent problem (Bistarelli, Montanari, & Rossi 1995) or that defined an NP-hard problem (Schiex, Fargier, & Verfaillie 1995). The absence of termination and non equivalence of a direct enforcing algorithm is a natural consequence from the fact that the addition of any non trivial (i.e. non tautological) constraint in a constraint network with a strictly monotonic combination operator will modify the semantics of the problem (the distribution of preferences over all complete assignments).

A possible approach, used in (Schiex 2000), is to try compensate for the addition of any constraint to the network in order to preserve equivalence. As an example, we consider the binary weighted MAX-CSPs in figure 1(a). This corresponds to valued CSPs using the strictly monotonic valuation structure $\langle \mathbb{N} \cup \{\infty\}, +, \geq \rangle$. To describe such problems, we use an undirected graph representation where vertices represent values. For all pairs of variables $i, j \in X$ such that $c_{ij} \in C$, for all values $a \in d_i$, $b \in d_j$ such that $c_{ij}(a, b) \neq \bot = 0$, an edge connect the values (i, a) and (j, b). The weight of this edge is set to $c_{ij}(a, b)$. Unary constraints are represented by weights associated with vertices, weights equal to 0 being omitted.

Our constraint network has two variables numbered 1 and 2, each with two values a and b together with a single constraint. The constraint forbids pair $((1, b), (2, b))$ with cost 1 and forbids pairs $((1, a), (2, a))$ and $((1, b), (2, a))$ completely (with cost ∞). The pair $((1, a), (2, b))$ is completely authorized and the corresponding edge is therefore omitted.

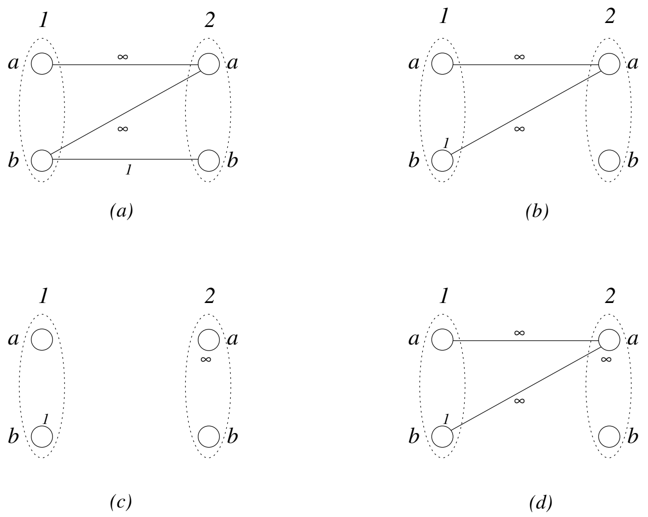

Figure 1: Four equivalent instances of MAX-CSP

If we assign the value b to variable 1, it is known for sure that a cost of 1 must be paid since all extensions of $(1, b)$ to variable 2 incur a cost of at least 1. Projecting this minimum cost down from c_{12} would make this explicit and induce a unary constraint on 1 that forbids $(1, b)$ with cost 1. However if we simply add this constraint to the MAX-CSP, as was proposed in (Bistarelli, Montanari, & Rossi 1995) for problems with an idempotent operator, the resulting CSP is not equivalent. The complete assignment $((1, b), (2, b))$ which initially had a cost of 1 would now have a cost of 2. In order to preserve equivalence, we must "compensate" for the induced unary constraint. This can be done by simply subtracting 1 from all the tuples that contain the value $(1, b)$. The corresponding equivalent CSP is shown in figure 1(b): the edge $((1, b), (2, b))$ of cost 1 has disappeared (the associated weight is now 0) while the edge $((1,b),(2,a))$ is unaffected since it has infinite weight. We can repeat this process for variable 2: all extensions of value $(2,a)$ have infinite cost. Thus we can add a unary constraint that completely forbids value $(2,a)$. In this specific case, and because the valuation ∞ satisfies $\infty \oplus \infty = \infty$, we can either compensate for this (Figure 1(c)) or not (Figure 1(d)). In both cases, an equivalent MAX-CSP is obtained. Between the problems in Figure 1(c)) and 1(d), we prefer the problem in Figure 1(d) because it makes information explicit both at the domain and constraint level.

This example leads to the following new axiom. This axiom effectively makes it possible to extend the arc consistency notion and algorithms as it is proved in (Schiex 2000; Cooper & Schiex 2002).

DEFINITION 6 *In a valuation structure $S = \langle E, \oplus, \succcurlyeq \rangle$, if $\alpha, \beta \in E$, $\alpha \preccurlyeq \beta$ and there exists a valuation $\gamma \in E$ such that $\alpha \oplus \gamma = \beta$, then γ is known as a difference of β and α.*

The valuation structure S is fair *if for any pair of valuations $\alpha, \beta \in E$, with $\alpha \preccurlyeq \beta$, there exists a maximal difference of β and α. This unique maximal difference of β and α is denoted by $\beta \ominus \alpha$.*

Actually, it has been proven that most usual valuation structure are fair. This includes possibilistic/min–max fuzzy CSP, weighted Max-CSP for example. These are either idempotent or strictly monotonic structures. Other in-between cases may occur which shows the relative generality of the VCSP framework.

DEFINITION 7 *In a valuation structure $\langle E, \oplus, \succcurlyeq \rangle$, an element $\alpha \in E$ is said to be an absorbing element iff $\alpha \oplus \alpha = \alpha$.*

The importance of these so-called "absorbing valuations" lies in the fact that they can be propagated as in classical CSP: the addition of such a valuation to a VCSP that already "express" it will not change it semantics. In a valuation structure $\langle E, \oplus, \succcurlyeq \rangle$, if $\oplus$ is idempotent then all elements of E are absorbing. This explains why arc consistency was straightforward to extend to idempotent structures. If $\oplus$ is a strictly monotonic operator then the only absorbing elements are $\perp$ and $\top$. Intermediate cases occur in the following examples:

EXAMPLE 1 *Imagine the possible sentences for driving offenses. Suppose that penalty points (up to a maximum of 12) are awarded for minor offenses, whereas serious offenses are penalized by suspension of the offender's driving license for a period of y years, for some positive integer y. A driver who accumulates 12 penalty points receives an automatic one-year suspension of his/her license. The set of sentences can be modeled by a valuation structure $S = \langle E, \oplus, \succcurlyeq \rangle$ of the form:*

$$E = \{(p,0) : p \in \{0,\dots,12\}\} \cup \{(0,y) : y \in \mathbb{N}^* \cup \{\infty\}\}$$

$$(p,y) \prec (p',y') \Leftrightarrow (y < y') \vee ((y = y' = 0) \wedge (p < p'))$$

$$\begin{aligned} (p,0) \oplus (p',0) &= (\min(p+p',12),0) \\ (p,y) \oplus (p',y') &= (0, y+y') \qquad \textit{if } (y+y' \neq 0) \end{aligned}$$

Note that $(12,0) \prec (0,1)$ even though they both give rise to a one-year license suspension. The penalty $(0,1)$ is deemed to be worse because it can be cumulated. For example $(0,1) \oplus (0,1) = (0,2)$, whereas $(12,0) \oplus (12,0) = (12,0)$. Apart from $\perp = (0,0)$ and $\top = (0,\infty)$, this valuation structure contains another absorbing valuation, namely $(12,0)$.

EXAMPLE 2 *Another interesting case occurs if, for example, a company wants to minimise both financial loss F and loss of human life H if a fire should break out in its factory. Supposing that the company considers that no price can be put on human life, we must have*

$$(F,H) < (F',H') \Leftrightarrow (H < H') \vee (H = H' \wedge F < F')$$

If a financial loss of $F_{\max}$ represents bankruptcy, then

$$(F,H) \oplus (F',H') = (\min\{F+F', F_{\max}\}, H+H')$$

and $(F_{\max},0)$ is an absorbing element which is strictly less than $\top$. Note that this valuation structure is not fair, since it is impossible to define $\alpha = (0,1) \ominus (F_{\max},0)$ such that $\alpha \oplus (F_{\max},0) = (0,1)$.

EXAMPLE 3 *Consider a valuation structure $S = \langle \mathbb{N} \cup \{\infty, \top\}, \oplus, \geq \rangle$ composed of prison sentences. Sentences may be of n years, life imprisonment (represented by ∞) or the death penalty (represented by $\top$). There is a rule that states that two life sentences lead automatically to a death sentence: in other words $(\infty \oplus \infty) = \top$. Otherwise, sentences are cumulated in the obvious way: $\forall m,n \in \mathbb{N}, (m \oplus n = m+n)$; $\forall n \in \mathbb{N}, (\infty + n = \infty)$; $\forall \alpha \in E, (\top \oplus \alpha = \top)$. Although every pair $\beta, \alpha \in E, \alpha \leq \beta$ possesses a difference, this valuation structure is not fair since the set of differences of ∞ and ∞ is $\mathbb{N}$ and hence no* maximal *difference of ∞ and ∞ exists. However, S can easily be rendered fair by replacing $\mathbb{N}$ by $\{0,1,2,\dots,150\}$, for example.*

These examples shows that fair valuation structures can capture preferences structures more complex than the usual additive (Shapiro & Haralick 1981), max (Rosenfeld, Hummel, & Zucker 1976), probabilistic (Fargier *et al.* 1995) or lexicographic (Fargier, Lang, & Schiex 1993) structures. A fine analysis of the general structure of fair valuation structures appears in (Cooper & Schiex 2002) where the following theorem is proved. It shows that in the most general case, a fair valuation structure is composed of "independent slices" separated by absorbing elements.

THEOREM 8 (SLICE INDEPENDENCE THEOREM) *Let $S = \langle E, \oplus, \succcurlyeq \rangle$ be a fair valuation structure. Let $\beta, \gamma \in E, \beta \preccurlyeq \gamma$, and let $\alpha_0, \alpha_1 \in E$ be absorbing valuations such that $\alpha_0 \preccurlyeq \gamma \preccurlyeq \alpha_1$. Then $\alpha_0 \preccurlyeq (\gamma \oplus \beta) \preccurlyeq \alpha_1$ and $\alpha_0 \preccurlyeq (\gamma \ominus \beta) \preccurlyeq \alpha_1$.*

Arc consistency in fair valued CSP

In classical CSPs, arc consistency enforcing always increases the information available on variables (by pruning values) and constraints (by implicitly removing tuples that use pruned values). In the case of soft arc consistency, in order to guarantee termination, this must be limited to operations that either increase the information available at the variable level or that increase information available at the constraint level *as long as they do not lower the information available at the variable level*. The latter will only be possible when the valuation propagated is absorbing.

The following result, proved in (Cooper & Schiex 2002) shows how absorbing elements that separate slices can be easily located.

THEOREM 9 *Let $S = \langle E, \oplus, \succcurlyeq \rangle$ be a fair valuation structure. For all $\alpha \in E$, $\alpha \ominus \alpha$ is the maximal absorbing valuation less than or equal to α.*

It os now possible to define arc consistency on all fair valuation structures. Note that this definition refines the definition of (Schiex 2000).

DEFINITION 10 *A fair binary VCSP is* arc consistent *if for all $i, j \in X$ such that $c_{ij} \in C^+$, for all $a \in d_i$ we have:*

1. $\forall b \in d_j, c_{ij}(a, b) = \big(c_i(a) \oplus c_{ij}(a, b) \oplus c_j(b)\big) \ominus \big(c_i(a) \oplus c_j(b)\big)$.
2. $c_i(a) = \min_{b \in d_j}(c_i(a) \oplus c_{ij}(a, b))$

Condition 1 states that $c_{ij}(a, b)$ has been increased to the maximal element in E which does not increase the valuation $\big(c_i(a) \oplus c_{ij}(a, b) \oplus c_j(b)\big)$ of (a, b) on $\{i, j\}$. If $\oplus$ is strictly monotonic or idempotent, then this is equivalent to saying that absorbing valuations have been propagated from $c_i(a)$ to $c_{ij}(a, b)$. Condition 2 says that we have propagated as much weight as possible from the constraint c_{ij} onto c_i.

To gain a better understanding of condition 1 of Definition 10 in the most general case, consider a simple valuation structure in which penalties lies in the range $\{0, 1, 2, 3, 4, 5\}$ and $\forall \alpha, \beta \in E, (\alpha \oplus \beta = \min(5, \alpha + \beta))$. 5 is absorbing and verifies $5 \ominus \alpha = 5$ for all $\alpha \preccurlyeq 5$. Figure 2(a) shows a 2-variable VCSP over this valuation structure. Figure 2(b) shows the result of enforcing condition 1 of Definition 10: $c_{12}(a, a)$ and $c_{12}(b, a)$ can both be increased to 5 without changing the valuations of the solutions (a, a) and (b, a). Figure 2(c) shows the result of then enforcing condition 2: penalties are projected down from constraints to domains, as we have seen in the example of Figure 1.

It is shown in (Cooper & Schiex 2002) that arc consistency can always be enforced in polynomial time on all fair valuation structures. All the usual properties of arc consistency (termination, in polynomial time, semantics preservation) but one are preserved: confluence of arc consistency enforcing is lost and therefore the arc consistent closure of a problem is not necessarily unique as it is in classical CSPs. Figure 3(a) shows a 2-variable VCSP on the valuation structure $\langle \mathbb{N} \cup \{\infty\}, +, \geq \rangle$. Each edge has a weight of 1. Figures 3(b) and 3(c) show two different arc consistency closures of this VCSP.

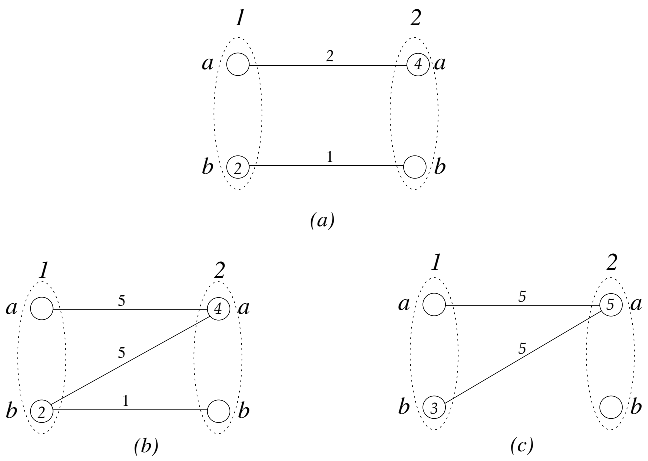

Figure 2: (a) An example of a VCSP and how conditions 1 and 2 are enforced

.

When the operator $\oplus$ is strictly monotonic, a specialized $O(ed^2)$ time, $O(ed)$ space enforcing algorithm can be defined.

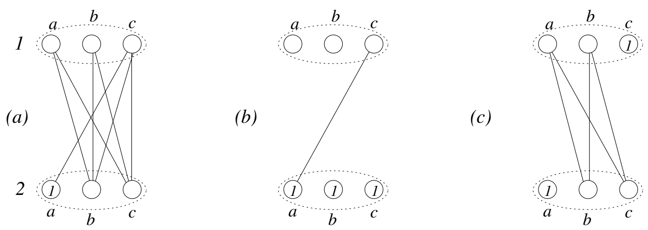

Figure 3: A MAX-CSP and two different equivalent arc consistent closures

From semi-groups to groups

Since the essential side-effect of the "fairness" axiom is the existence of a difference between valuations, it seems reasonable to consider using a group-based structure instead of a semi-group based structure.

Intuitively, this seems extremely attractive since it will allow to express both dislikes and preferences that can compensate with each other. Naturally, since a difference will always be defined in such a structure, one could hope for the existence of arc consistency enforcing algorithms. Furthermore, opposite are much easier to handle than differences in practice and would probably lead to simpler proofs and algorithms.

Such a structure can be defined as follows:

DEFINITION 11 *A* symmetric valuation structure *is defined as a tuple $\langle E, \oplus, \ominus, \succcurlyeq \rangle$ such that:*

- *E is a set, whose elements are called valuations, which is totally ordered by $\succcurlyeq$, with a maximum element denoted by $\top$.*

- *E is closed under a commutative, associative binary operation $\oplus$ that satisfies:*
  - Identity*: $\exists 0 \in E, \forall \alpha \in E, \alpha \oplus 0 = \alpha$;*
  - Monotonicity*: $\forall \alpha, \beta, \gamma \in E, (\alpha \succcurlyeq \beta) \Rightarrow \big((\alpha \oplus \gamma) \succcurlyeq (\beta \oplus \gamma)\big)$;*
  - Maximum*: $\forall \alpha \in E, (\alpha \oplus \top) = \top$.*
  - Opposite*: $\forall \alpha \in E - \{\{\top\}, \exists! \beta \in E, \alpha \oplus \beta = 0$. Such a β is noted $\ominus\alpha$.*

Note that this set of axioms does not entail the set of axioms of valuation structures. Indeed, the element 0 is no more minimum. It is important to show that this set of axioms is not inconsistent (or we could prove many properties on such structures), but quite obviously, the classical boolean structure of classical CSP can easily be cast as such a symmetric valuation structure. We get $E = \{\top, 0\}$ with $\top > 0$ and $\oplus = \max$.

Keeping all other definitions essentially identical (definition of a symmetric valued CSP and of the valuation of a tuple in a symmetric valued CSP), a first important remark is that we loose an essential property: the valuation of an assignment is not necessarily a bound of the valuation of the complete assignment. However, such a trivial lower bound is not used anymore in practice because of its poor quality.

So, this loss does not remove interest in symmetric valuation structure. We can either develop dedicated lower bounds or we can restrict ourself to express problems with only "positive" valuations. In this case, we obviously recover the fact that the valuation of a local assignment is a lower bound on the valuation of a complete assignment. Such a restriction could be interesting if it allows to extend arc consistency to a significant set of structures.

The following result shows that all strictly monotonic valuations structure can essentially be "remapped" to a symmetric valuation structure. This result (and its proof) are very reminiscent of the closely related result that all strictly monotonic valuation can be embedded in a fair valuation structure (Cooper 2002).

PROPERTY 12 *From any strictly monotonic valuation structure $V = \langle E, \oplus, \succcurlyeq \rangle$, we can build a symmetric valuation $V' = \langle E', \oplus', \ominus, \succcurlyeq' \rangle$ structure and a morphism $f : E \to E'$ such that $f(a \oplus b) = f(a) \oplus' f(b)$ and $a \succ b \Leftrightarrow f(a) \succ' f(b)$.*

Proof: Consider $F = \{(a,b) \mid a \in E, b \in E - \{\{\top\}\}$ and the relation $\equiv$ on $F \times F$ defined by $(a,b) \equiv (c,d) \Leftrightarrow (a \oplus d) = (b \oplus c)$. The fact that $\equiv$ is an equivalence relation follows from the strict monotonicity of $\oplus$: $\forall (a,b),(c,d),(e,f) \in E \times E - \{\top\}$: we have $(a \oplus d = b \oplus c) \wedge (c \oplus f = d \oplus e) \Rightarrow a \oplus d \oplus c \oplus f = b \oplus c \oplus d \oplus e \Rightarrow a \oplus f = b \oplus e$ if $(c \oplus d) \neq \top$. Otherwise, if $(c \oplus d) = \top$, this implies that either $c = \top$ or $d = \top$ and therefore $a = e = \top$ so that $a \oplus f = b \oplus e$.

Let F' be the set of equivalence classes of $\equiv$ in F. If we identify $a \in E$ with the equivalence class of $(a, \bot)$ in F', we see that F' is an extension of E. We write 0 and $\top'$ for (the equivalence classes of) $(\bot, \bot)$ and $(\top, \bot)$. We define $\succcurlyeq'$ by $(a,b) \succcurlyeq' (c,d) \Leftrightarrow a \oplus d \succcurlyeq b \oplus c$. It is easy to see that $\succcurlyeq'$ is well-defined. The aggregation $\oplus'$ is defined as $(a,b) \oplus' (c,d) = (a \oplus c, b \oplus d)$. The opposite is defined as $\ominus(a,b) = (b,a)$ for $(a,b) \in F' - \{\top'\}$.

That $\succcurlyeq'$ is a total order follows from the fact that $(a,b) \succcurlyeq' (c,d) \wedge (c,d) \succcurlyeq' (a,b) \Rightarrow a \oplus d = b \oplus c \Rightarrow (a,b) \equiv (c,d)$ and $(a,b) \succcurlyeq' (c,d) \wedge (c,d) \succcurlyeq' (e,f) \Rightarrow (a \oplus d \succcurlyeq b \oplus c) \wedge (c \oplus f \succcurlyeq d \oplus e)$. This implies that $(a \oplus d \oplus c \oplus f \succcurlyeq b \oplus c \oplus d \oplus e \wedge c \neq \top) \vee (a = c = \top)$. Therefore $a \oplus f \succcurlyeq b \oplus e$ (by strict monotonicity) and $(a,b) \succcurlyeq' (e,f)$.

The fact that F' is closed under $\oplus$ follows from the definition of $\oplus$. That $\top'$ is the maximum element of F' follows from the fact that $\forall (a,b) \in F', b \oplus \top \succcurlyeq a \oplus \bot$ and therefore $(\top, \bot) \succcurlyeq' (a,b)$. That 0 is the identity of F' follows from the fact that $\forall (a,b) \in F', (a,b) \oplus' (\bot, \bot) = (a \oplus \bot, b \oplus \bot) = (a,b)$.

The fact that the $\ominus(a,b)$ is the opposite of (a,b) follows from the fact that $\forall (a,b) \in F', (\bot \oplus (a \oplus b) = \bot \oplus (a \oplus b)$ which means that $(\bot, \bot) \equiv (a \oplus b, a \oplus b) \equiv (a,b) \oplus' (b,a) = (a,b) \oplus' \ominus(a,b)$.

The commutativity and associativity of $\oplus'$ in F' follow directly form the associativity and commutativity of $\oplus$ in E. Strict monotonicity follows from the fact that $(a,b),(c,d),(e,f) \in F' \wedge (a,b) \succ' (c,d) \wedge (e,f) \not\equiv \top' \Rightarrow a \oplus d \succ b \oplus c \wedge \top \neq e \succcurlyeq f \Rightarrow a \oplus e \oplus d \oplus f \succ b \oplus f \oplus c \oplus e \Rightarrow (a \oplus e, b \oplus f) \succ' (c \oplus e, d \oplus f) \Rightarrow (a,b) \oplus' (e,f) \succ' (c,d) \oplus' (e,f)$. The absorbing element axiom follows from the fact that $\forall (a,b) \in F', (a,b) \oplus' \top' = (a \oplus \top, b \oplus \bot) = (\top, b) \equiv \top'$.

The morphism f from E to F' is simply defined by $\forall a \in E, f(a)$ is the equivalence class of $(a, \bot)$ in F'. Very simply, $\forall a,b \in E, f(a \oplus b) \equiv (a \oplus b, \bot) = (a, \bot) \oplus' (b, \bot) \equiv f(a) \oplus' f(b)$. Furthermore $a \succ b \Leftrightarrow (a \oplus \bot \succ b \oplus \bot) \Leftrightarrow (a, \bot) \succ' (b, \bot) \Leftrightarrow f(a) \succ' f(b)$. □

This shows that from any strictly monotonic VCSP, we can map it to a symmetric VCSP by mapping all valuations in it using the f morphism. The corresponding problem can be processed and solved and the optimum solution will be an optimal solution of the original problem.

The main point that remains to be addressed is to see if we can find morphisms between other (non strictly monotonic) valuation structures and symmetric valuation structures.

PROPERTY 13 *No such morphism exists between a non strictly monotonic valuation structure and any symmetric valuation structure.*

Proof: If we consider a non strictly monotonicity valuation structure $V = \langle E, \oplus, \succ \rangle$, this means that $\exists a,b,c \in E, c \neq \top$ such that $a \succ b$ and $a \oplus c = b \oplus c$.

Imagine that a morphism exists between V and a symmetric valuation structure $V' = \langle E', \oplus', \ominus, \succcurlyeq' \rangle$. We then have $f(a \oplus c) = f(b \oplus c) \Leftrightarrow f(a) \oplus' f(c) = f(b) \oplus' f(c)$. By existence of an opposite number this implies that $f(a) \oplus' f(c) \oplus' \ominus f(c) = f(b) \oplus' f(c) \oplus' \ominus f(c) \Leftrightarrow f(a) = f(b)$ which contradicts the fact that $a \succ b$. □

As a reviewer pointed out, an alternative way of reaching this result is to show that the combination of the axioms of triangular co-norms which appear in the valuation structure axioms (associativity, commutativity, absorbing element, identity and monotonicity) with the additional property of being extensible to a domain E' such that each element of E has an opposite in the extended domain E' implies strict monotonicity. This shows that the possible morphims are restricted because the possible symmetric valuation structures are actually restricted to strictly monotonic structures.

So, it appears that the introduction of an opposite in valuation structures that could ease the definition of arc consistency enforcing algorithms essentially restricts the framework to strictly monotonic structures which an important loss of expressivity. Although one could argue that strictly monotonic structures covers important cases (eg/ Max-CSP), we consider that such an refinement leads to a too specific scheme. Specifically, it will not cover instances as described in Figure 2 that corresponds to the situation of branch and bound: since an upper bound on the cost of the optimal solution is known, all solutions whose valuations are equal to or larger than this upper bound are considered as completely uninteresting and should therefore be immediately pruned.

Conclusion

Following our work on arc consistency, we thought that a possible evolutions of the valuation structure axioms could be to introduce the existence of an opposite as an improvement of the artificial and relatively complex fairness axiom. As we shown, This replacement actually consists in shifting from a semi-group based structure to a group based structure. We have shown that this actually essentially limits the structures captured to strictly monotonic structures which is a relatively heavy cost to pay.

Further work is now needed to see how fair valuation structures could be weakened to capture partial orders without loosing the existence of polynomial time arc consistency enforcing algorithms.

References

Aho, A.; Hopcroft, J.; and Ullman, J. 1974. *The Design & Analysis of Computer Algorithms*. Addison-Wesley.

Bertelé, U., and Brioshi, F. 1972. *Nonserial Dynamic Programming*. Academic Press.

Bessière, C. 1991. Algorithmes d'arc-consistance pour les problèmes de satisfaction de contraintes dynamiques. Technical Report 91-086, LIRMM Montpellier II, Montpellier, France.

Bistarelli, S.; Fargier, H.; Montanari, U.; Rossi, F.; Schiex, T.; and Verfaillie, G. 1999. Semiring-based CSPs and valued CSPs: Frameworks, properties and comparison. *Constraints* 4:199–240.

Bistarelli, S.; Montanari, U.; and Rossi, F. 1995. Constraint solving over semirings. In *Proc. of the 14^{th} IJCAI*.

Cooper, M., and Schiex, T. 2002. Arc consistency for soft constraints. *submitted to JACM*. see arXiv.org/abs/cs.AI/0111038.

Cooper, M. C. 2002. Reduction operations in fuzzy or valued constraint satisfaction. *Fuzzy Sets and Systems*. To appear.

Dechter, R. 1999. Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence* 113(1–2):41–85.

Dubois, D., and Prade, H. 1982. A class of fuzzy measures based on triangular norms. a general framework for the combination of uncertain information. *Int. Journal of Intelligent Systems* 8(1):43–61.

Fargier, H.; Lang, J.; Martin-Clouaire, R.; and Schiex, T. 1995. A constraint satisfaction framework for decision under uncertainty. In *Proc. of the 11^{th} Int. Conf. on Uncertainty in Artificial Intelligence*.

Fargier, H.; Lang, J.; and Schiex, T. 1993. Selecting preferred solutions in Fuzzy Constraint Satisfaction Problems. In *Proc. of the 1^{st} European Congress on Fuzzy and Intelligent Technologies*.

Freuder, E. C. 1989. Partial constraint satisfaction. In *Proc. of the 11^{th} IJCAI*, 278–283.

Larrosa, J.; Meseguer, P.; and Schiex, T. 1999. Maintaining reversible DAC for Max-CSP. *Artificial Intelligence* 107(1):149–163.

Minoux, M. 1976. Structures algébriques généralisées des problèmes de cheminement dans les graphes. *Recherche opérationnelle* 10(6):33–62.

Rosenfeld, A.; Hummel, R.; and Zucker, S. 1976. Scene labeling by relaxation operations. *IEEE Trans. on Systems, Man, and Cybernetics* 6(6):173–184.

Schiex, T.; Fargier, H.; and Verfaillie, G. 1995. Valued constraint satisfaction problems: hard and easy problems. In *Proc. of the 14^{th} IJCAI*, 631–637.

Schiex, T. 1992. Possibilistic constraint satisfaction problems or "How to handle soft constraints ?". In *Proc. of the 8^{th} Int. Conf. on Uncertainty in Artificial Intelligence*.

Schiex, T. 2000. Arc consistency for soft constraints. In *Principles and Practice of Constraint Programming - CP 2000*, volume 1894 of *LNCS*, 411–424.

Shapiro, L., and Haralick, R. 1981. Structural descriptions and inexact matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 3:504–519.

Shenoy, P. 1991. Valuation-based systems for discrete optimization. In Bonissone; Henrion; Kanal; and Lemmer., eds., *Uncertainty in AI*. North-Holland Publishers.