Constraints, 10, 283–305, 2005

High-Order Consistency in Valued Constraint Satisfaction

MARTIN C. COOPER cooper@irit.fr
IRIT, University of Toulouse III, 118 route de Narbonne, 31062 Toulouse, France

Abstract. *k*-consistency operations in constraint satisfaction problems (CSPs) render constraints more explicit by solving size-*k* subproblems and projecting the information thus obtained down to low-order constraints. We generalise this notion of *k*-consistency to valued constraint satisfaction problems (VCSPs) and show that it can be established in polynomial time when penalties lie in a discrete valuation structure.

A generic definition of consistency is given which can be tailored to particular applications. As an example, a version of high-order consistency (face consistency) is presented which can be established in low-order polynomial time given certain restrictions on the valuation structure and the form of the constraint graph.

Keywords: discrete optimisation, valued constraint satisfaction, soft constraints, consistency enforcing, MAX-CSP

1. Valued Constraint Satisfaction

The Valued Constraint Satisfaction Problem (VCSP) is a generalisation of the classic CSP in which the user can express preferences between satisfying assignments in under-constrained problems or between near-miss assignments in over-constrained problems. Preferences are expressed by specifying penalties on assignments to subsets of the variables. In Mathematical Programming penalties traditionally lie in $\Re \cup \{\infty\}$. However, a great diversity of problems can be modelled by varying the range of possible penalties and the operator which is used to aggregate them. The notion of valuation structure [20] captures the minimal set of properties that a set of penalties must satisfy.

Definition 1.1 A valuation structure is a tuple $\langle E, \oplus, \geq \rangle$ such that

- E is a set, whose elements are called valuations, which is totally ordered by $\geq$, with a maximum element denoted by $\top$ and a minimum element denoted by $\bot$;
- E is closed under a binary operation $\oplus$ that satisfies:
  - $\forall \alpha, \beta \in E, \alpha \oplus \beta = \beta \oplus \alpha$; (*commutativity*)
  - $\forall \alpha, \beta, \gamma \in E, \alpha \oplus (\beta \oplus \gamma) = (\alpha \oplus \beta) \oplus \gamma$; (*associativity*)
  - $\forall \alpha, \beta, \gamma \in E, \alpha \geq \beta \Rightarrow (\alpha \oplus \gamma) \geq (\beta \oplus \gamma)$; (*monotonicity*)
  - $\forall \alpha \in E, \alpha \oplus \bot = \alpha$; (*neutral element*)
  - $\forall \alpha \in E, \alpha \oplus \top = \top$. (*annihilator*)

A valuation structure is a positive totally ordered commutative monoid, also known as a positive tomonoid. Note that in the semi-ring based constraint satisfaction problem

(SCSP), valuations satisfy the same properties except that they are only partially ordered [2].

Definition 1.2 An operator $\oplus$ is idempotent if $\forall \alpha \in E$, $\alpha \oplus \alpha = \alpha$. It is strictly monotonic if $\forall \alpha, \beta, \gamma \in E$, $(\alpha > \beta) \wedge (\gamma \neq \top) \Rightarrow \alpha \oplus \gamma > \beta \oplus \gamma$.

Note that the only possible idempotent operator in a valuation structure is max [2]. Examples of strictly monotonic aggregation operators include addition in the non-negative reals with infinity, multiset union in the case of the leximin version of the Fuzzy CSP [14] and the operator $p \oplus q = 1 - (1 - p)(1 - q)$ in the probabilistic CSP [13]. We call a valuation structure idempotent (strictly monotonic) if its operator is idempotent (strictly monotonic).

Define the addition-with-ceiling operator $+_m$ as follows:

$$\forall a, b \in \{0, 1, \ldots, m\} \quad a +_m b = \min\{\mathrm{a} + \mathrm{b}, m\}$$

Then for $m > 1$, $S_m = \langle \{0,1,\ldots,m\}, +_m, \geq \rangle$ is a valuation structure in which $+_m$ is neither idempotent nor strictly monotonic, since $(m - 1) < (m - 1) +_m (m - 1) = m = \top = m +_m (m - 1)$. In a bounded version of MAX-CSP (known as B-MAX-CSP in this paper), penalties lie in the valuation structure S_m [16]. It is a version of MAX-CSP in which all solutions which violate m or more constraints are considered equally bad. This is a situation which applies, for example, at a node of a branch and bound search tree on a MAX-CSP problem where m is the number of constraints violated by the best solution found so far.

Definition 1.3 In a valuation structure $\langle \mathrm{E}, \oplus, \geq \rangle$, $\alpha \in E$ is an *idempotent valuation* if $\alpha \oplus \alpha = \alpha$.

In the conjunctive version of the Fuzzy CSP [18] all valuations are idempotent. If the aggregation operator $\oplus$ is strictly monotonic, then the only idempotent valuations are $\bot$ and $\top$. Idempotent valuations can always be propagated since this does not alter the valuation of any solution.

Definition 1.4 [9] In a valuation structure $\langle E, \oplus, \geq \rangle$, if $\alpha, \beta \in E$, $\alpha \leq \beta$ 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 is fair if for any pair of valuations $\alpha, \beta \in E$, with $\alpha \leq \beta$, there exists a maximum difference of β and α. This unique maximum difference of β and α is denoted by $\beta \ominus \alpha$.

For example, if $\oplus$ is addition in $\Re^+ \cup \infty$, then $\ominus$ is subtraction of real numbers (with $\infty \ominus \infty = \infty$). If $\oplus$ is max, then $\ominus$ is also max [19]. If $\oplus$ is $+_m$, then $\ominus$ is $-_m$ given by $\forall \alpha, \beta \in \{0, 1, \ldots, m\}$ such that $m > \alpha \geq \beta$, $\alpha -_m \beta = a - b$ and $\forall \beta \in \{0, 1, \ldots, m\}$, $m - \beta = m$ [16, 17].

Definition 1.5 [9] A *Valued Constraint Satisfaction Problem* (VCSP) is a tuple $\langle N, D, C, S\rangle$ where N is a set of n variables $N = \{1, \ldots, n\}$, each variable $i \in N$ has a domain of possible values $d_i \in D$, C is a set of constraints and $S = \langle E, \oplus, \geq\rangle$ is a valuation structure. Each constraint in C is defined over a set of variables $P \subseteq N$ (its scope) as a function ϕ_P from the cartesian product of the domains $d_i (i \in P)$ to E.

Purely for notational convenience, we suppose that no two constraints have the same scope. This allows us to identify C with the set of scopes P of constraints ϕ_P in the VCSP.

Notation For $P \subseteq N$ we denote the cartesian product of the domains d_i $(i \in P)$ (i.e., the set of possible labellings for the variables in P) by $L(P)$. It is important that $L(P)$ be non-empty even when P is the empty set. Thus $L(\emptyset) = \{()\}$ where $()$ represents the labelling of 0 variables.

Notation Let $P \subseteq Q \subseteq N$ with $Q = \{i_1, \ldots, i_q\}$ and $P = \{j_1, \ldots, j_p\}$. Then, given an assignment $t = \left(t_{i_1}, \ldots t_{i_q}\right) \in L(Q)$, $\pi_P t$ denotes the subassignment of t to the variables in P, i.e., $\left(t_{j_1}, \ldots, t_{j_p}\right)$.

Definition 1.6 In a VCSP $V = \langle N, D, C, S\rangle$, the *valuation* of an assignment $t \in L(N)$ is defined by

$$Val_V(t) = \bigoplus_{P \in C} [\phi_P(\pi_P t)]$$

To solve a VCSP we have to find an assignment $t \in L(N)$ with a minimum valuation.

A CSP can be viewed as a VCSP over the idempotent valuation structure $\langle\{0, \infty\}, +, \geq\rangle$. MAX-CSP, the problem of maximising the number of satisfied constraints in an over-constrained CSP, is a VCSP over the valuation structure $\langle \mathbf{N} \cup \{\infty\}, +, \geq\rangle$ in which the constraint functions can only take on the values 0 or 1. We denote by ∞-MAX-CSP the equivalent problem in which constraint functions can take on values 0,1 or ∞ (thus allowing the possibility of completely inconsistent tuples). B-MAX-CSP (Bounded MAX-CSP) is MAX-CSP over the valuation structure S_m.

1.1. Some Basic Definitions and Properties of VCSPs

The following theorem, proved in the Appendix, effectively shows that we need never consider valuation structures with idempotent valuations other than $\bot$ and $\top$.

Theorem 1.7 *Any fair VCSP V can be solved by (1) solving a single conjunctive Fuzzy CSP, and (2) solving a single VCSP over a valuation structure with only two idempotent valuations* $\bot$, $\top$.

It is well known that a conjunctive Fuzzy CSP can be solved by binary search in its valuation structure, which consists in solving a logarithmic number of CSPs [6].

Definition 1.8 A valuation structure $\langle E, \oplus, \geq \rangle$ is discrete if for each $\alpha \in E$ such that $\alpha < \top$ there is a finite number of $\beta \in E$ such that $\beta \leq \alpha$.

MAX-CSP, ∞-MAX-CSP and B-MAX-CSP have discrete valuation structures. If $V = \langle N, D, C, S \rangle$ is a VCSP over a discrete valuation structure S, then there are only a finite number of distinct VCSPs V' which are equivalent to V. This is because the constraint functions of V' can only take on valuations in $\{\beta : \exists t \in L(N)(\beta \leq Val_V(t) < \top)\} \cup \{\top\}$, of which there are only a finite number since S is discrete. It is this property of discreteness which provides a guarantee that the consistency-enforcing algorithms described in this paper terminate. Note that neither $\langle \Re^+ \cup \{\infty\}, +, \geq \rangle$ nor the valuation structure of the leximin version of the FCSP [14] are discrete. The following theorem, proved in the Appendix, provides a characterisation of discrete fair valuation structures.

Theorem 1.9 *If a valuation structure $S = \langle E, \oplus, \geq \rangle$ is discrete, fair and has only two idempotent valuations $\bot$, $\top$, then it is isomorphic to either $\mathbf{N} \cup \{\infty\}$ or S_m for some $m \in \mathbf{N}$.*

Theorems 1.7 and 1.9 tell us that if we restrict ourselves to discrete fair valuation structures, then we need only provide consistency algorithms for VCSPs over the valuations structures $\mathbf{N} \cup \{\infty\}$ and S_m.

Definition 1.10 Two VCSPs $V_1 = \langle N, D, C_1, S \rangle$, $V_2 = \langle N, D, C_2, S \rangle$ are equivalent if $\forall t \in L(N) Val_{V_1}(t) = Val_{V_2}(t)$.

Definition 1.11 The subproblem of a VCSP $\langle N, D, C, S \rangle$ on $Q \subseteq N$ is the problem VCSP(Q) = $\langle Q, D_Q, C_Q, S \rangle$, where $D_Q = \{d_i : i \in Q\}$ and $C_Q = \{P \in C : P \subseteq Q\}$.

Definition 1.12 For a VCSP $\langle N, D, C, S \rangle$, an equivalence preserving transformation (*EPT*) on $Q \subseteq N$ is an operation which transforms the subproblem VCSP(Q) into an equivalent VCSP.

Definition 1.13 If ϕ_P, ϕ_Q are two constraint functions with scopes P, Q (respectively), then their sum $\phi = \phi_P \oplus \phi_Q$ is the constraint function with scope $P \cup Q$ such that

$$\forall t \in L(P \cup Q) \quad \phi(t) = \phi_P(\pi_P t) \oplus \phi_Q(\pi_Q t)$$

If $Q \subseteq P$ then the difference $\psi = \phi_P \ominus \phi_Q$ has scope P and is given by

$$\forall t \in L(P) \quad \psi(t) = \phi_P(t) \ominus \phi_Q(\pi_Q t)$$

Definition 1.14 If ϕ_J is a constraint function with scope J and $I \subseteq J$, then the projection $\psi = \mathrm{PROJ}_I \phi_J$ of ϕ_J onto I is the constraint function with scope I such that

$$\forall t \in L(I) \quad \psi(t) = \mathrm{MIN}\{\phi_J(z) : z \in L(J) \wedge \pi_I z = t\}$$

Note that when I is the empty set, the projection ψ of ϕ_J onto I is the nullary constraint given by the constant $\psi = \text{MIN}\{\phi_J(z) : z \in L(J)\}$.

Definition 1.15 A VCSP is known as binary if none of its constraint scopes has cardinality greater than 2.

Definition 1.16 If V is VCSP, then its constraint graph has a node for each variable and an edge joining each pair of distinct nodes i, j for which there is a constraint ϕ_P in V with $i, j \in P$.

Definition 1.17 If $P, Q \subseteq N$, then $t \in L(P)$ is compatible with $y \in L(Q)$ if $\pi_{P\cap Q}\, t = \pi_{P\cap Q}\, y$.

Note that, if $P \cap Q = \emptyset$, then every $t \in L(P)$ is compatible with every $y \in L(Q)$.
The following properties of valuation structures were proved in [9].

Lemma 1.18 *Let $S = \langle E, \oplus, \geq\rangle$ be a fair valuation structure and let $\alpha, \beta \in E$. Then*

(a) *$\alpha \ominus \alpha$ is the maximum idempotent valuation less than or equal to α.*
(b) *if either α or β is idempotent, then $\alpha \oplus \beta = \max\{\alpha, \beta\}$.*

Lemma 1.19 *Let $S = \langle E, \oplus, \geq\rangle$ be a fair valuation structure and let $\alpha, \beta \in E$. Then either $(\alpha \oplus \beta) \ominus \beta = \alpha$ or $(\alpha \oplus \beta) \ominus \beta = (\alpha \oplus \beta) \ominus (\alpha \oplus \beta)$ which is idempotent and strictly greater then α.*

The following two lemmas are essential for the proof of correctness of the consistency algorithms presented in this paper.

Lemma 1.20 *Let $S = \langle E, \oplus, \geq\rangle$ be a fair valuation structure and let $\alpha, \beta, \gamma \in E$. If $\beta \oplus \gamma \leq \alpha$ then $\alpha \ominus (\beta \oplus \gamma) \leq \alpha \ominus \beta$.*

Proof: By definition of $\ominus$, we have $\alpha = (\alpha \ominus (\beta \oplus \gamma)) \oplus (\beta \oplus \gamma) = ((\alpha \ominus (\beta \oplus \gamma)) \oplus \gamma \oplus \beta$. By maximality of $\alpha \ominus \beta$, we can deduce that $(\alpha \ominus (\beta \oplus \gamma)) \oplus \gamma \leq \alpha \ominus \beta$. Thus, by monotonicity, $\alpha \ominus (\beta \oplus \gamma) \leq (\alpha \ominus (\beta \oplus \gamma)) \oplus \gamma \leq \alpha \ominus \beta$. ■

Lemma 1.21 *Let $S = \langle E, \oplus, \geq\rangle$ be a fair valuation structure and let $\mu, \mu', \rho \in E$. If $\mu' \leq \mu$, then $(\mu' \oplus \rho) \ominus \mu' \leq (\mu \oplus \rho) \ominus \mu$.*

Proof: By Lemma 1.19, either (i) $(\mu' \oplus \rho) \ominus \mu' = \rho$ or (ii) $(\mu' \oplus \rho) \ominus \mu' = (\mu' \oplus \rho) \ominus (\mu' \oplus \rho)$ which is idempotent and strictly greater than ρ. Now, again by Lemma 1.19, $(\mu \oplus \rho) \ominus \mu \geq \rho$. Thus, in case (i), $(\mu' \oplus \rho) \ominus \mu' \leq (\mu \oplus \rho) \ominus \mu$ and we are done. In case (ii), $(\mu' \oplus \rho) \ominus \mu' = (\mu' \oplus \rho) \ominus (\mu' \oplus \rho) \leq (\mu \oplus \rho) \ominus (\mu \oplus \rho)$ (by Lemma 1.18 (a) since $\mu' \oplus \rho \leq \mu \oplus \rho$ by monotonicity). But $(\mu \oplus \rho) \ominus (\mu \oplus \rho) \leq (\mu \oplus \rho) \ominus \mu$ (by Lemma 1.20 with $\alpha = \mu \oplus \rho$, $\beta = \mu$, $\gamma = \rho$). ■

2. High-Order Consistency in VCSPs

Given the importance of different forms of consistency in CSPs, a natural goal is to generalise the notion of consistency to valued constraints. The different forms of consistency which have already been defined for VCSPs include arc consistency [9, 19], (full) directional arc consistency [6] and cyclic consistency [7].

Arc consistency not only propagates total inconsistencies (as in CSPs), but also projects down weights from binary to unary constraints. Directional arc consistency attempts to concentrate weights on the same variables by shifting weights towards earlier variables in a given variable-ordering. Full directional arc consistency (FDAC) is simply arc consistency and directional arc consistency applied simultaneously. Maintaining FDAC during branch and bound search was found to be the most efficient pruning technique based on arc consistency operations in experimental trials on hard instances of MAX-CSP [17]. Cyclic consistency re-assigns weights on binary and unary constraints around a cycle of variables. In binary MAX-CSP, it is known that an order-3 cyclic consistency closure is in-scope optimal, in the sense that the lower bound on valuations of solutions thus obtained cannot be locally improved by any form of order-3 consistency operation which does not create ternary constraints [7].

In this section, we give a generic definition of consistency in VCSPs which allows us to define complete k-consistency, where $k = 3$ is any positive integer. In particular, when $k = 3$ this provides a form of 3-consistency which is strictly stronger than order-3 cyclic consistency at the cost of extra space complexity, since all ternary constraints may be created. (Note, however, that it is possible to avoid explicitly storing order-3 constraints when the valuation structure is strictly monotonic [8], based on an idea already applied to arc consistency in VCSPs [9]). Our generic definition of consistency also allows us to define a restricted form of higher-order consistency which can be applied in low-order polynomial time on certain forms of planar constraint graphs.

Notation Let C be a set of scopes (i.e., subsets of N). Then maximal (C) denotes the set of elements of C which are maximal with respect to subset inclusion, i.e., maximal (C) = $\{J \in C : \nexists K \in C \text{ such that } J \subset K\}$.

Notation $Val_J^{\leq i}$ is the valuation function of the subproblem of VCSP(J) consisting of constraints of order no greater than i, i.e.,

$$\forall z \in L(J) \quad Val_J^{\leq i}(z) = \bigoplus_{P \in C \wedge P \subseteq J \wedge |P| \leq i} \phi_P(\pi_P z)$$

$Val_J^{>i}$ is defined similarly and Val_J is simply the valuation function of the subproblem VCSP(J):

$$\forall z \in L(J) \quad Val_J(z) = \bigoplus_{P \in C \wedge P \subseteq J} \phi_P(\pi_P z)$$

Consistency in a fair VCSP V corresponds to a state in which, after solving a given set of subproblems of V, idempotent valuations have been propagated as much as possible

and all valuations have been projected down to lower-order constraints. In particular, complete k-consistency can be established by solving all size-k subproblems, propagating any resulting idempotent valuations, projecting down weights onto subproblems of these size-k subproblems and repeating these operations until convergence. We give a generic definition of consistency as a function of a set C of scopes (which in complete k-consistency will simply be all sets of at most k variables).

Definition 2.1 A fair VCSP is C-consistent if $\forall J \in \text{maximal}(C), \forall I \in C$ such that $I \subseteq J$,

$$\phi_I = \phi_I \oplus \text{PROJ}_I\left(Val_J \ominus Val_J^{\leq |I|}\right)$$

Definition 2.2 A fair *VCSP* is complete k-consistent if it is C-consistent for $C = \{J \subseteq N : |J| \leq k\}$.

Definition 2.2 differs from the definition of k-consistency in CSPs [3] and the definition of arc consistency in VCSPs given in previous papers [6, 9, 19] in that there is a constraint with an empty scope (since C includes all subsets of size less than or equal to k, including the empty set). A constraint with an empty scope, known as the nullary constraint and denoted simply by ϕ, has proved useful in arc consistency [16], soft MAC [17] and cyclic consistency enforcing [7]. Our previous definition of arc consistency [9] in VCSPs corresponds therefore to C-consistency with $C = \{J \subseteq N : 1 \leq |J| \leq 2\}$.

If $\oplus$ is idempotent, then $Val_J \ominus Val_J^{\leq |I|} = Val_J$ and $\phi_I = \phi_I \oplus \text{PROJ}_I Val_J$ iff $\phi_I = \text{PROJ}_I Val_J$. Thus, in CSPs, complete k-consistency imposes the classic notion of strong k-consistency which says that $\forall I \subset J \subseteq N$ such that $|J| = k$, any consistent labelling for I can be extended to a consistent labelling for J. Definition 2.2 also imposes the condition that inconsistent labellings for J be eliminated (whereas strong k-consistency does not, since it is not necessary to explicitly store order-k constraints). Note that, when $\oplus$ is not idempotent, an increase in ϕ_I is compensated by a decrease in ϕ_J and hence order-k constraints must be stored.

If $\oplus$ is addition of real numbers and all valuations are finite (as is the case in MAX-CSP), then $Val_J \ominus Val_J^{\leq |I|} = Val_J^{> |I|}$, the valuation function of the subproblem of VCSP(J) involving only constraints of order greater than $|I|$, and $\phi_I = \phi_I \oplus \text{PROJ}_I Val_J^{> |I|}$ iff $\text{PROJ}_I Val_J^{> |I|} = 0$. This corresponds to a situation in which all labellings t for I can be extended to a labelling for J at no extra cost in terms of higher-order constraints.

To establish complete k-consistency in VCSPs we propagate non-idempotent weights from ϕ_P to ϕ_J (where $P \subset J$) only if this can lead to an increase in ϕ_I for some $I \subset J$ such that $|I| < |P|$. Since $|I| < |P| < |J|$, this means that it is impossible to check complete k-consistency by tests on pairs of constraint functions ϕ_I, ϕ_J such that $|J| = |I| + 1$, as is the case in CSPs (in which complete k-consistency is equivalent to i-consistency for $1 \leq i \leq k$).

As an example of complete 3-consistency, consider the VCSP V shown in Figure 1(a). Each oval represents a variable domain. In fact, each domain is Boolean and is

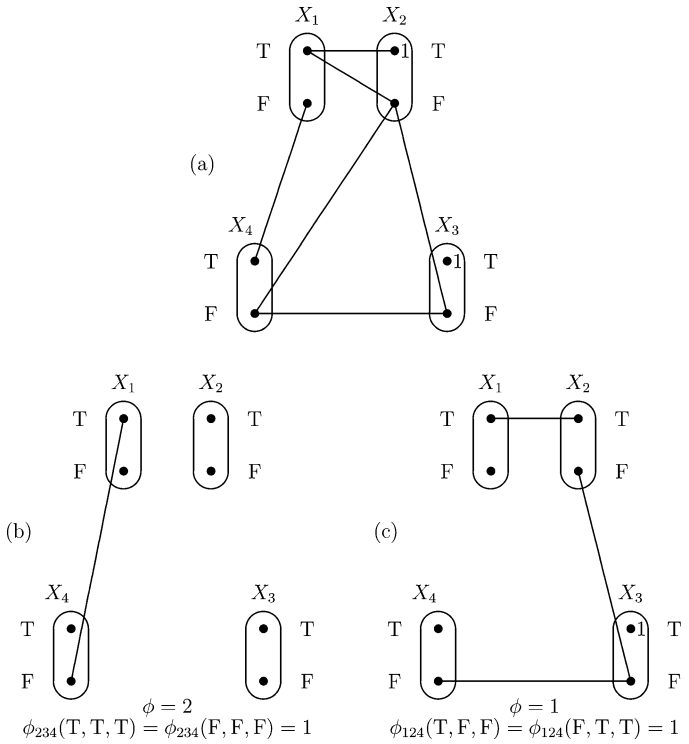

Figure 1. (a) A VCSP V; (b), (c) two distinct complete 3-consistency closures of V.

represented in Figure 1 by {T, F}. A number 1 next to a value a in the domain of X_i represents a cost of 1 (i.e., $\phi_i(a) = 1$). Each line joining a in the domain of X_i with b in the domain of X_j represents a cost of 1 (i.e., $\phi_{ij}(a, b) = 1$). V corresponds to an instance of MAX-SAT in which the aim is to satisfy the maximum number of the following constraints: $\neg X_1 \vee \neg X_2$; $\neg X_1 \vee X_2$; $\neg X_2$; $\neg X_3$; $X_2 \vee X_3$; $X_1 \vee \neg X_4$; $X_2 \vee X_4$; $X_3 \vee X_4$. V is not complete 3-consistent: for example, if ψ_2 denotes $\mathrm{PROJ}_{\{2\}}\ Val^{>1}_{\{1,2,4\}} = \mathrm{PROJ}_{\{2\}}(\phi_{12} \oplus \phi_{14} \oplus \phi_{24})$, then $\psi_2(F) = 1 > 0$. Note that $Val^{>1}_{\{1,2,4\}}$ is the sum of all constraints ϕ_P such that $P \subseteq \{1,2,4\} \wedge |P| > 1$, including $P = \{1, 4\}$ which does not actually intersect $\{2\}$.

When the operator $\oplus$ is not idempotent, the complete k-consistency closure is not necessarily unique. To illustrate this, consider the VCSP V of Figure 1. Figures 1(b) and (c) show two distinct complete 3-consistency closures of V. In both cases the resulting VCSP consists not only of the unary and binary constraints shown diagrammatically but also the nullary constraint ϕ and a single ternary constraint (for which we only indicate the non-zero penalties).

The valuation ϕ is a natural lower bound on the valuation of any solution. Indeed, producing a tight lower bound for use in branch and bound search is the principal reason for applying consistency operations. For example, between the two complete 3-con-

sistency closures given in Figure 1(b) and (c), the former is preferable to the latter since the lower bound ϕ is greater. Note that finding a complete k-consistency closure for which ϕ is maximal is known to be NP-hard in the case of MAX-CSP, even for $k = 2$ [9].

The generic definition of consistency in Definition 2.1 allows us to define restricted forms of higher-order consistency which are more time and space efficient than complete k-consistency. If the constraint graph G of a binary VCSP is planar, then an obvious choice for the set of scopes C is the set of nodes, edges and faces of G.

Definition 2.3 A fair binary VCSP V whose constraint graph G is planar is face consistent if V is C-consistent for $C = \{\emptyset\} \cup \{\{v\} : v \in N\} \cup E \cup F$, where N, E, F are respectively the nodes, edges and faces of G (where a face is understood to mean the set of nodes on its boundary) and $\emptyset$ is the empty set.

Imagine the problem of colouring the nodes of a planar graph G such that the minimum number of pairs of adjacent nodes are assigned the same colour. This is an optimisation version of the well-known graph-colouring problem. Suppose that only two colours (black and white) are available and that G is the graph shown in Figure 2(a). An optimal colouring is obtained, for example, by colouring all nodes on the same bottom-left-to-top-right diagonal the same colour, with alternating black and white diagonals (as indicated in Figure 2(a)). In the face-consistency closure illustrated in Figure 2(b), the value of the nullary constraint ϕ is 4 which is, in fact, the valuation of an optimal solution. In this case, face-consistency finds the lower bound of 1 for the number of pairs of adjacent nodes assigned the same colour in each of the 4 shaded triangles in Figure 2(b). In each shaded triangle, the three "not-equal-to" constraints (given by $\phi_{12}(\mathrm{B,W}) = \phi_{12}(\mathrm{W,B}) = 0$; $\phi_{12}(\mathrm{B,B}) = \phi_{12}(\mathrm{W,W}) = 1$) are replaced by a nullary constraint $\phi = 1$ and a ternary "not-all-equal" constraint (given by $\phi_{123}(\mathrm{B,B,B}) = \phi_{123}(\mathrm{W,W,W}) = 2$; $\phi_{123}(x, y, z) = 0$ otherwise).

We should mention that other face-consistency closures exist for the problem in Figure 2(a). One example is illustrated in Figure 2(c) where $\phi = 2$, the number of shaded triangles. In all face-consistency closures, $2 \leq \phi \leq 4$. On a general $2n \times 2n$ triangular grid, the lower bound ϕ found by face consistency (if the triangles are visited in the natural left-to-right top-to-bottom order) is $4n^2$, which is the value of an optimal solution. The lower bound found by face consistency is at worst $2n^2$, which is obtained only if the triangles are visited in a rather unnatural order. The corresponding face consistency closures on a $2n \times 2n$ grid can be represented by tesselating the plane with the patterns of Figures 2(b) and (c).

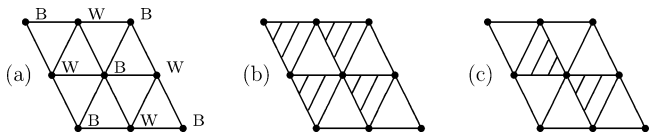

Figure 2. (a) A graph G to be coloured with just two colours (an optimal colouring is shown); (b), (c) after applying face-consistency ϕ is the number of shaded triangles.

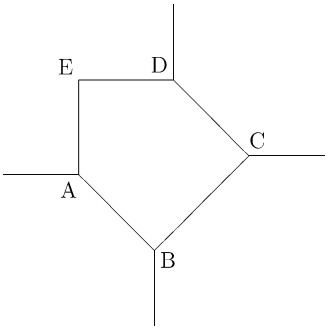

Figure 3. A fragment of a line drawing labelling problem.

It is worth noting that both full directional arc consistency [6, 9] and 3-cyclic consistency [7] are completely ineffective on this problem instance since they both produce a lower bound of $\phi = 0$. For this particular problem, face consistency not only produces an excellent lower bound, it can also clearly be established in linear time.

As an example of face consistency on a classical CSP, consider the fragment of a line drawing of a polyhedra in Figure 3. The line drawing labelling problem can be expressed as a CSP with a variable X_j corresponding to each junction j in the drawing. The domain of X_j is the list of legal labellings of the lines meeting at j. For example, one legal labelling of a Y-junction is (+++) representing three convex edges (such as the corner of a cube from a viewpoint at which all three faces which intersect at the corner are visible). On the other hand, the labelling (++) is illegal for an L-junction. There is a binary constraint between each pair of adjacent junctions j, k: the line joining junctions j and k must be assigned the same label in the labellings X_j and X_k. Under the assumptions of a general viewpoint and trihedral vertices, a Y-junction has no legal labelling involving a convex edge other than the labelling (+++). It is easy to deduce that the labelling (+++) for junction A in Figure 3 would imply that junctions B, C, D should also be labelled (+++). But then this is a contradiction, since the labelling (++) is illegal for the L-junction E. Face consistency therefore tells us that (+++) is an illegal labelling for A.

3. Enforcing Consistency

In this section we present a generic algorithm **GC** to establish C-consistency.

Definition 3.1 For $I, J \in C$, such that $I \subseteq J$, and for $t \in L(P)$, a VCSP is said to be C-consistent at (I, t, J) if $\phi_I(t) = \phi_I(t) \oplus \text{PROJ}_I(Val_J \ominus Val_{\widetilde{J}}^{\leq |J|})(t)$.

The algorithm **GC**, given in Figure 4, keeps a list L of triples (I, t, J) which need to be checked for C-consistency. At any moment during the execution of **GC** the VCSP is C-consistent except possibly at those $(I, t, J) \in L$.

GC:
L := $\{(I,t,J) : J \in \text{maximal}(C) \wedge I \in C \wedge I \subseteq J \wedge t \in L(I)\}$;
while $L \neq \emptyset$ do
(*Invariant INV: VCSP is C-consistent except possibly at $(I,t,J) \in L$ *)
Extract (I,t,J) from L ;
δ := **PValDiff**(I,t,J) ;
if $\delta \oplus \phi_I(t) > \phi_I(t)$ then (* C-inconsistent at (I,t,J) *)
Re-establish(I,t,J,δ) ; (* C-consistent at (I,t,J) but need to recheck consistency on all $J' \in \text{maximal}(C)$ such that $I \subseteq J'$ *)
Add $\{(I',t',J')$:
$I, I' \subseteq J' \wedge I' \in C \wedge J' \in \text{maximal}(C) \wedge t' \in L(I')\}$ to L ;
end if ;
end while ;
(* PostCondition: VCSP is C-consistent *)

Figure 4. A generic algorithm to establish C-consistency.

PValDiff(I, t, J) returns the value of $\text{PROJ}_I(Val_J \ominus Val_J^{\leq |I|})$ at t. When an inconsistency is detected, C-consistency is re-established by the subroutine **Re-establish**, given in Figure 5, which first shifts weights from constraint functions ϕ_P (for $|P| > |I|$) up to to ϕ_J, and then projects down weights to ϕ_I. Remember that $L(P)$ represents the set of labellings for P and that $y \in L(P)$ is said to be compatible with $t \in L(I)$ iff y and t agree on variables $P \cap I$.

Extend and **Propagate** both shift penalties up to the constraint ϕ_J so that a penalty of δ can then be projected down from ϕ_J onto $\phi_I(t)$. Optimisations of **GC** are possible. For example, it is not necessary to call **Extend** $(P, y, J, \min(\phi_P(y), \delta))$ in **Re-establish** (I, t, J, δ) if $\phi_P(y) \oplus \phi_I(t) = \phi_I(t)$. (In MAX-CSP this occurs if $\phi_P(y) = 0$). As another example, in MAX-CSP it is not necessary to call **Propagate** (since the only idempotent valuation is 0) and we need only add those (I', t', J') to L in **GC** which also satisfy $(|I'| < |I|) \vee (I' \subset J' = J)$ (since ϕ_I and ϕ_J are the only constraint functions which may have increased).

Consider the execution of **GC** on the VCSP given in Figure 1(a) in order to establish complete 3-consistency. In this case, C is the set of all subsets of $\{X_1, X_2, X_3, X_4\}$ and $\text{maximal}(C) = \{\{X_1, X_2, X_3\}, \{X_1, X_2, X_4\}, \{X_1, X_3, X_3\}, \{X_2, X_3, X_4\}\}$. Suppose that, after execution of the first line of **GC**, the list L begins as follows $L = \{(\emptyset,()), \{X_1, X_2, X_3\}), (\{X_1\}, (\text{T}), \{X_1, X_2, X_3\}), \ldots\}$. Then, during the first iteration of the while loop, $(I, t, J) = (\emptyset, (), \{X_1, X_2, X_3\})$. Since $\text{MIN}\{Val_J(z) : z \in L(J)\} = 1$ and the initial value of the nullary constraint is 0, we have

$$\delta = \textbf{PValDiff}(\emptyset, (), \{X_1, X_2, X_3\}) = 1$$

Now clearly $1 \oplus 0 > 0$ (the nullary constraint being equal to 0). Thus **GC** has detected an inconsistency. **Re-establish** $(\emptyset, (), \{X_1, X_2, X_3\}, 1)$ first calls **Extend** which shifts all weights up to ϕ_{123} from all ϕ_P for $P \subset \{X_1, X_2, X_3\}$ and $|P| > 0$. **Re-establish** then calls **Project** which assigns 1 to the nullary constraint ϕ and reduces $\phi_{123}(z)$ by 1 for all

```
Re-establish(I, t, J, δ):
  if δ is an idempotent valuation then φ_I(t) := φ_I(t) ⊕ δ ; return ;
  end if ;
  for all P ⊂ J such that P ∈ C ∧ |P| > |I| do
      for all y ∈ L(P) compatible with t ∈ L(I) do
          (* Shift weights up to φ_J *)
          Extend(P, y, J, min(φ_P(y), δ)) ;
      end for ;
  end for ;
  (* Propagate idempotent valuations up to φ_J *)
  Propagate(I, t, J) ;
  (* Project weight of δ from φ_J down to φ_I(t) *)
  Project(I, t, J, δ) ;


Extend(P, y, J, δ):
  for all w ∈ L(J) compatible with y ∈ L(P) do
      φ_J(w)  :=  φ_J(w) ⊕ δ ;
  end for ;
  φ_P(y) := φ_P(y) ⊖ δ ;

Propagate(I, t, J):
  for all z ∈ L(J) compatible with t ∈ L(I) do
      ρ := Val_J(z) ;
      φ_J(z) := φ_J(z) ⊕ (ρ ⊖ ρ) ;
  end for ;

Project(I, t, J, δ):
  for all z ∈ L(J) compatible with t ∈ L(I) do
      φ_J(z)  :=  φ_J(z) ⊖ δ ;
  end for ;
  φ_I(t) := φ_I(t) ⊕ δ ;
```

Figure 5. **Re-establish** and its subroutines.

$z \in L(\{X_1, X_2, X_3\})$. Note that we now have $\phi = 1$ but $\phi_P(y) = 0$ for all P such that $\emptyset \subset P \subset J$ and all $y \in L(P)$. The resulting VCSP is illustrated in Figure 6(a).

On subsequent iterations, some of the weights now stored in ϕ_{123} will be projected down to constraint functions ϕ_P where $P \subset \{X_1, X_2, X_3\}$. For example, when $(I, t, J) = (\{X_1\}, (\mathrm{T}), \{X_1, X_2, X_3\})$, **GC** calculates

$$\begin{aligned}\delta &= \textbf{PValDiff}(\{X_1\}, (\mathrm{T}), \{X_1, X_2, X_3\}) \\ &= \mathrm{PROJ}_{\{X_1\}}\left(Val_{\{X_1,X_2,X_3\}} \ominus Val^{\leq 1}_{\{X_1,X_2,X_3\}}\right)(\mathrm{T}) \\ &= \min\{1, 1, 1, 2\} = 1\end{aligned}$$

and thus detects another inconsistency. **Re-establish** calls **Extend** to shift all weights up to ϕ_{123} from all $\phi_P(y)$ such that $P \subset \{X_1, X_2, X_3\}$, $|P| > 1$ and y compatible with $X_1 = \mathrm{T}$.

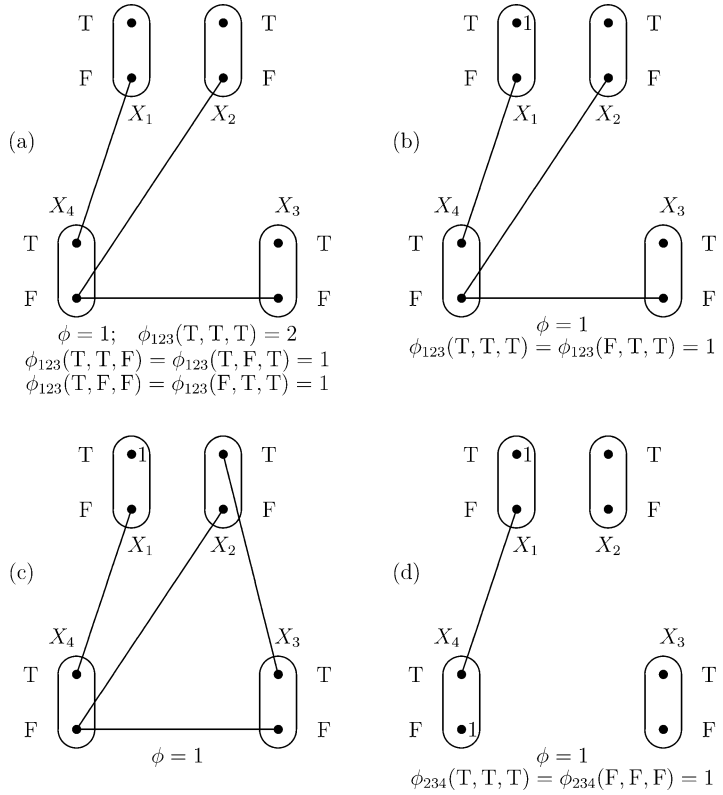

Figure 6. The intermediate results of applying **GC** to the VCSP of Figure 1 (a), after re-establishing consistency at (a) (I, t, J) = ($\emptyset$, ( ), $\{X_1, X_2, X_3\}$); (b) (I, t, J) = ($\{X_1\}$, (T), $\{X_1, X_2, X_3\}$); (c) (I, t, J) = ($\{X_2, X_3\}$, (T,T), $\{X_1, X_2, X_3,\}$); (d) (I, t, J) = ($\{X_4\}$, (F), $\{X_2, X_3, X_4\}$).

In fact, there are no non-zero weights to be shifted up. **Re-establish** then calls **Project** which assigns 1 to $\phi_1(\mathrm{T})$ and reduces $\phi_{123}(z)$ by 1 for all z such that $\pi_{X_1} z = T$. The resulting VCSP is given in Figure 6(b). In a subsequent iteration of the while loop, with (I, t, J) = ($\{X_2, X_3\}$, (T, T),$\{X_1, X_2, X_3\}$), **Re-establish** projects down a weight of 1 from ϕ_{123} onto ϕ_{23}(T, T). The result is shown in Figure 6(c).

The next iteration of the while loop of **GC** which detects an inconsistency occurs when (I, t, J) = ($\{X_4\}$, (F), $\{X_1, X_2, X_3\}$). The result of **Re-establish** ($\{X_4\}$, (F), $\{X_1$, $X_2, X_3\}$, 1) is shown in Figure 6(d). Note that all the non-zero weights on the three binary constraints ϕ_{23}, ϕ_{34}, ϕ_{24} must be shifted up to ϕ_{234} in order to be able to project a weight of 1 down to $\phi_4(\mathrm{F})$. A final inconsistency is detected by **GC** when (I, t, J) = ($\emptyset$, ( ), $\{X_1$, $X_4\}$) since, for all labellings (t_1, t_4) of $\{X_1, X_4\}$, $\phi_1(t_1) \oplus \phi_4(t_4) \oplus \phi_{1,4}(t_1, t_4) \geq 1$. The complete 3-consistency closure shown in Figure 1(b) is then obtained by applying **Re-establish** ($\emptyset$, ( ), $\{X_1, X_4\}$, 1).

3.1. *Correctness of the Consistency-Enforcing Algorithm*

In order to show that **GC** is correct we have to show that it is an EPT (equivalence-preserving transformation) and that on termination the resulting VCSP is C-consistent. We first show that each of the subroutines of **GC** are EPTs.

Lemma 3.2 Propagate(I, t, J) *is an EPT on VCSP*(J) *if* $J \in C$, $t \in L(I)$ *and* $I \subseteq J$.

Proof: Consider some $z \in L(J)$ compatible with $t \in L(I)$ and let ρ represent the value of $Val_J(z)$ before the call of **Propagate**(I, t, J). Then after this call of **Propagate**, $Val_J(z) = \rho \oplus (\rho \ominus \rho) = \rho$ and hence is invariant. ■

Lemma 3.3 Project (I, t, J, δ) *is an EPT on VCSP*(J) *provided that* $I, J \in C$, $t \in L(I)$, $I \subseteq J$ *and* $\delta \leq \text{MIN}\{\phi_J(z) : z \in L(J)$ *compatible with* $t \in L(I)\}$.

Proof: The result follows from the fact that, for all $z \in L(J)$ compatible with $t \in L(I)$,

$$\phi_J(z) \oplus \phi_I(\pi_I z) = \phi_J(z) \oplus \phi_I(t)$$

is an invariant of **Project**(I, t, J, δ) since $(\phi_J(z) \ominus \delta) \oplus (\phi_I(t) \oplus \delta) = ((\phi_J(z) \ominus \delta) \oplus \delta) \oplus \phi_I(t) = \phi_J(z) \oplus \phi_I(t)$, and hence $Val_J(z)$ is also an invariant. If, on the other hand, $z \in L(J)$ is not compatible with $t \in L(I)$, then clearly **Project** (I, t, J, δ) does not modify $Val_J(z)$. ■

Lemma 3.4 Extend(P, y, J, δ) *is an EPT on VCSP*(J) *if* $P, J \in C$, $y \in L(P)$, $P \subset J$ *and* $\delta \leq \phi_P(y)$.

Proof: As in the proof of Lemma 3.3, this follows from the fact that $\forall w \in L(J)$, $\phi_J(w) \oplus \phi_P(\pi_P w) = \phi_J(w) \oplus \phi_P(y)$ is an invariant of **Extend**(P, y, J, δ), and hence $Val_J(w)$ is also an invariant. ■

Lemma 3.5 Re-establish(I, t, J, δ) *is an EPT on VCSP*(J) *when* $I, J \in C$, $t \in L(I)$, $I \subseteq J$ *and*

$$\delta = \min\left\{Val_J(z) \ominus Val_J^{\leq |I|}(z) : z \in L(J) \text{ compatible with } t \in L(I)\right\}$$

Proof: First of all, consider the case in which δ is idempotent. In this case, **Re-establish**(I, t, J, δ) simply adds δ to $\phi_I(t)$. But $\forall z \in L(J)$ compatible with $t \in L(I)$, $Val_J(z) \ominus Val_J^{\leq |I|}(z) \geq \delta$. Therefore $Val_J(z) \geq \delta \oplus Val_J^{\leq |I|}(z) \geq \delta$ by monotonicity. Therefore $Val_J(z) \oplus Val_J(z)$ since δ is idempotent, by Lemma 1.18(b). Thus, when δ is idempotent, adding δ to $\phi_I(t)$ leaves $Val_J(z)$ invariant for all $z \in L(J)$ and hence **Re-establish**(I, t, J, δ) is an EPT on VCSP(J).

Now consider the case in which δ is not idempotent. We can assume that $I \subset J$, since if $I = J$ then $\delta = Val_J(z) \ominus Val_J(z)$, for some $z \in L(J)$, which is idempotent by Lemma 1.18(a).

Knowing that **Propagate**, **Extend** and **Project** are EPTs, to show that **Re-establish** is also an EPT it only remains to show that whenever **Project**(I, t, J, δ) is called the condition of Lemma 3.3 is verified, namely

$$\forall z \in L(J) \text{ compatible with } t \in L(I), \phi_J(z) \geq \delta \tag{1}$$

Consider some $z \in L(J)$ compatible with $t \in L(I)$. Since $I \subset J$, this implies that $\pi_1 z = t$. At the end of the calls to **Extend** in **Re-establish**(I, t, J, δ), $\phi_J(z) = Val0_J^{>|I|}(z)$, where $Val0$ represents the valuations of the VCSP before the call to **Re-establish**(I, t, J, δ).

Now, by the definition of δ in the statement of the lemma, $\delta \leq Val0_J(z) \ominus Val0_J^{\leq|I|}(z)$. Setting $\mu = Val0_J^{>|I|}(z)$ and $\lambda = Val0_J^{\leq|I|}(z)$, we can write this as $\delta \leq (\mu \oplus \lambda) \ominus \lambda$. By Lemma 1.19, either

(i) $(\mu \oplus \lambda) \ominus \lambda = \mu$ or
(ii) $(\mu \oplus \lambda) \ominus \lambda = (\mu \oplus \lambda) \ominus (\mu \oplus \lambda)$

By monotonicity, $\mu \oplus ((\mu \oplus \lambda) \ominus (\mu \oplus \lambda)) \geq \max\{\mu, (\mu \oplus \lambda) \ominus (\mu \oplus \lambda)\} \geq (\mu \oplus \lambda) \ominus \lambda$ in both case (i) and case (ii). After the call of **Propagate**(I, t, J), $\phi_J(z) = \mu \oplus (Val0_J(z) \ominus Val0_J(z)) = \mu \oplus ((\mu \oplus \lambda) \ominus (\mu \oplus \lambda)) \geq (\mu \oplus \lambda) \ominus \lambda \geq \delta$. Thus we have shown that equation (1) holds when **Project**(I, t, J, δ) is called. ■

Having shown that **Re-establish** is an EPT, we now turn our attention to the invariant of the while loop in **GC**.

Lemma 3.6 *Suppose that* $P \subseteq J$, $H \subseteq K$, $P, H \in C$, $J, K \in \text{maximal}(C)$, $J \neq K$, $y \in L(P)$, $z \in L(H)$ *and* $\delta \leq \phi_P(y)$. *If the VCSP is C-consistent at* (H, z, K), *then it remains C-consistent at* (H, z, K) *after* **Extend**(P, y, J, δ) *is called.*

Proof: C-consistency on (H, z, K) depends only on the values of $\phi_Q(v)$ for $Q \subseteq K$ and $v \in L(Q)$ compatible with $z \in L(H)$. **Extend**(P, y, J, δ) only modifies ϕ_J and $\phi_P(y)$. Since $J \neq K$ and $J, K \in$ maximal (C), we cannot have $J \subseteq K$. We also assume that $P \subseteq K$ and $z \in L(H)$ is compatible with $y \in L(P)$, otherwise the result is trivial. Thus P, $H \subset K$ and $P \subset J$.

Suppose that the VCSP is C-consistent at (H, z, K) before the call of **Extend**(P, y, J, δ). Then there exists $w \in L(K)$ compatible with (i.e., an extension of) $z \in L(P)$ such that

$$\phi_H(z) = \phi_H(z) \oplus \left(Val_K \ominus Val_K^{\leq|H|}\right)(w) \tag{2}$$

Let ρ, ρ' be (respectively) the values of $Val_K^{>|H|}(w)$ before and after the call of **Extend**(P, y, J, δ). Clearly $\rho' \leq \rho$. (In fact, we can only have $\rho' \neq \rho$ if $|P| > |H|$). Let μ and μ' be (respectively) the values of $Val_K^{\leq|H|}(w)$ before and after **Extend**(P, y, J, δ) is called. Clearly $\mu' \leq \mu$. (In fact, we can only have $\mu' \neq \mu$ if $|P| \leq |H|$). Let ϕ and ϕ' be (respectively) the values of $\phi_H(z)$ before and after **Extend**(P, y, J, δ) is called. (In fact, we can only have $\phi \neq \phi'$ in the case $P = H$).

Equation (2) can be written as $\phi = \phi \oplus ((\mu \oplus \rho) \ominus \mu)$. From the definition of $\phi \ominus \phi$ as a maximal difference, this implies that $(\mu \oplus \rho) \ominus \mu \leq \phi \ominus \phi$. Since $\mu' \leq \mu$, Lemma 1.21 tells us that $(\mu' \oplus \rho) \ominus \mu' \leq (\mu \oplus \rho) \ominus \mu \leq \phi \ominus \phi$. Since $\rho' \leq \rho$, by monotonicity, $(\mu' \oplus \rho') \ominus \mu' \leq (\mu' \oplus \rho) \ominus \mu' \leq \phi \ominus \phi$. Now, if $P \neq H$ then $\phi' = \phi$ and if $P = H$ then, by the definition of **Extend**, $\phi' = (\phi \ominus \delta) \geq (\phi \ominus \phi)$ since by hypothesis $\delta \leq \phi_P(y) = \phi_H(z) = \phi$. From Lemma 1.18(a) it follows that $(\phi' \ominus \phi') \geq (\phi \ominus \phi) \geq (\mu' \oplus \rho') \ominus \mu'$. It follows from monotonicity that $\phi' = \phi' \oplus (\phi' \ominus \phi') \geq \phi' \oplus ((\mu' \oplus \rho') \ominus \mu') \geq \phi'$, and hence $\phi' = \phi' \oplus ((\mu' \oplus \rho') \ominus \mu')$. In other words, the VCSP is C-consistent at (H, z, K) (since equation (2) still holds) after the call of **Extend** (P, y, J, δ). ■

Theorem 3.7 *If* **GC** *terminates then it establishes C-consistency.*

Proof: To demonstrate partial correctness it is clearly sufficient to show that **GC** is an EPT and that the resulting VCSP is C-consistent. That **GC** is an EPT follows directly from Lemma 3.5. To show that the resulting VCSP is C-consistent, we only need prove that INV (which says that the VCSP is C-consistent except possibly at $(I, t, J) \in L$) is an invariant of the while loop. The only constraint functions modified during an iteration of the while loop in which (I, t, J) is processed are ϕ_I, ϕ_J and ϕ_P for $P \subset J$. Lemma 3.6 tells us that the reductions to ϕ_P (for $P \subset J$) by **Extend** (P, y, J, δ) cannot lead to C-inconsistencies. Thus **Re-establish** (I, t, J, δ) can only destroy C-consistency at those triples (I', t', J') such that $I, I' \subseteq J'$, $I' \in C$, $I \neq \emptyset$, $J' \in$ maximal (C) and $t' \in L(I')$. But these are exactly the tuples added to L after the call of **Re-establish** (I, t, J, δ), and hence INV is an invariant of the while loop. ■

Total correctness of **GC**, when the valuation structure of the VCSP is $\mathbf{N} \cup \{\infty\}$ or S_m (for $m \in \mathbf{N}$), follows from the analysis of its time complexity in the following section.

3.2. Complexity of Establishing Consistency

Theorem 3.8 *If k is a constant and $\forall I \in C$, $|I| \leq k$, then C-consistency can be established by* **GC** *in polynomial time on all instances of ∞-MAX-CSP and B-MAX-CSP.*

Proof: Consider first the execution of **GC** on an instance of MAX-CSP. Let N_i be the total number of times **Re-establish** (I, t, J, δ) is called for some I such that $|I| \leq i$. Since **Re-establish** (I, t, J, δ) clearly has polynomial time complexity, it is sufficient to show that N_k is polynomially-bounded.

The value of $\phi_I(t)$ necessarily increases each time **Re-establish** (I, t, J, δ) is called. Note that **Re-establish** (I, t, J, δ) may be called many times for the same pair (I, t), not least because $\phi_I(t)$ may also decrease (for $|I| > 0$).

For all $I \subseteq N$ and for all $t \in L(I)$, $\phi_I(t)$ is bounded above by the maximum valuation of any solution, which for MAX-CSP is c, the number of constraints. In particular, the nullary constraint ϕ is bounded above by c. Since ϕ is never decreased by **GC**, we can deduce that $N_0 \leq c$.

Re-establish (I, t, J, δ) increases $\phi_I(t)$ by δ, but may also decrease $\phi_P(y)$ by δ for all (y, p) such that $y \in L(P)$, $|I| < |P|$ and $P \subseteq J$: a total decrease which is nevertheless bounded above by $\sum_{p=1}^{k} C_k^p d^p \delta = (1+d)^k \delta$ (since $|J| \leq k$). Thus, for $0 < i \leq k$,

$$N_i \leq cd^i C_n^i + (1+d)^k N_{i-1}$$

since there are $d^i C_n^i$ values $\phi_P(y)$ (with $|P| \leq i$ and $y \in L(P)$) each bounded above by c and each increase in some $\phi_I(t)$ where $|I| < |P|$ may provoke at most $(1+d)^k$ decreases in some $\phi_P(y)$.

Since $N_0 \leq c$, it is easy to show by induction that

$$N_i \leq c\Big(dn + (1+d)^k\Big)^i$$

and hence that N_k is polynomially-bounded under the assumption that k is a constant.

When the VCSP is an instance of ∞-MAX-CSP or B-MAX-CSP, then the above argument can be used to show that N_k' is polynomially-bounded, where N_i' is the number of times **Re-establish** (I, t, J, δ) is called for some I such that $|I| \leq i$ with $\delta < \top$. The result follows from the fact that **Re-establish** (I, t, J, δ) is called at most once with $\delta = \top$ for each pair (I, t) since constraint values $\phi_I(t) = \top$ are never decreased by **GC**. ■

Theorem 3.9 *If k is a constant and $\forall J \in C$, $|J| \leq k$, then C-consistency can be established in polynomial time on all instances of VCSP over the valuation structures* $\mathrm{N} \cup \{\infty\}$ *and S_m (for $m \in \mathrm{N}$).*

Proof: Let I be a VCSP instance over the valuation structure $\mathbf{N} \cup \{\infty\}$, and let M be the maximum finite valuation taken on by the constraint functions in I. Define I_1 to be a copy of I in which each constraint function ϕ_J^1 in I_1 is related to the corresponding constraint function ϕ_J in I by the rule $\phi_J^1 = \phi_J \bmod 2$. The function mod is understood in its normal sense except that $\top \bmod 2$ is defined and is equal to $\top$.

Let $\overline{\phi}_J^1$ be the constraint functions of I_1 after establishment of C-consistency. Then $\overline{\phi}_J^1 \bmod 2$ is the least significant bit of a C-consistent VCSP which is equivalent to I. Now, define each I_i (for $i \geq 2$) to be a copy of I with constraint functions ϕ_J^i given by

$$\phi_J^i = ((\phi_J + \overline{\phi}_J^{i-1}) \text{ div } 2^{i-1}) \bmod 2$$

where $\overline{\phi}_J^{i-1}$ is the result of applying C-consistency to I_{i-1}. The function div is integer division, except that $\top \text{ div } 2^{i-1} = \top$. The function $\overline{\phi}_J^{i-1}$ is the "carry" from I_{i-1}. After establishing C-consistency in I_i, the resulting constraint functions are $\overline{\phi}_J^i$. The value $(\overline{\phi}_J^i \bmod 2)$ represents the ith least significant bit of the constraint functions of $\overline{I}$, a C-consistent VCSP equivalent to the original instance I. Note that each I_i is an instance of ∞-MAX-CSP or B-MAX-CSP since all valuations are either 0,1 or $\top$.

The maximum finite valuation of the constraint functions after establishing C-consistency is bounded by the sum of the finite valuations in the original VCSP I, which is at most cM. The number of iterations of the above procedure (i.e., the largest value of

i for which we need to calculate I_i) is thus $O(\log(cM))$. This is $O(\log c + b)$ where b is the maximum number of bits required to store the valuations in the original VCSP I.

The result follows directly from Theorem 3.8 for the case in which the valuation structure is $\mathbf{N} \cup \{\infty\}$. For a VCSP over the valuation structure S_m, we can use the above algorithm except that each time some $\phi_J(t)$ becomes $\top$ we restart the whole algorithm. The number of such restarts is bounded by cd^k. Hence the total complexity is again polynomial. ■

Corollary 3.10 *If k is a constant, then complete k-consistency can be established in polynomial time by* **GC** *on all instances of VCSP over the valuation structures* $\mathbf{N} \cup \{\infty\}$ *and S_m (for $m \in \mathbf{N}$).*

Note that the average-case time complexity of establishing consistency can be expected to be much less than the worst-case upper bound given in the proof of Theorem 3.8. Moreover, there is no obligation to apply consistency operations until complete convergence.

To demonstrate that practical versions of C-consistency exist, we give below a result which is proved in detail in [8], concerning the complexity of face consistency. We say that a graph is degree-bounded if the maximum node-dregree is a constant. This is the case, for example, in the line drawing labelling problem for drawings of objects with trihedral vertices [4, 5], where the maximum node-degree is 3.

Theorem 3.11 *In an instance of binary ∞-MAX-CSP on a planar degree-bounded constraint graph G, face consistency can be established in $O(nd^6k^2)$ time and $O(nd^2)$ space, where k is the maximum number of nodes on the boundary of a face.*

Again, we should note that $O(nd^6k^2)$ is a pessimistic worst-case upper bound on time complexity which is unlikely to be attained in practice. The $O(nd^2)$ space complexity comes from the fact that it is unnecessary to store the constraint functions ϕ_P for faces P. Instead we store the net sum of the valuations projected from ϕ_R to ϕ_P for all nodes and edges R, based on a technique already used in soft arc consistency enforcing [9]. The fact that the time complexity is linear in n is simply due to the fact that a planar graph of n nodes has $O(n)$ edges and faces. A naive algorithm would have an exponential time complexity in k, the maximum number of nodes per face, but, by using the fact that a face is a partial 2-tree, it is well known that this can be avoided [1, 12].

4. Discussion

In the special case of $k = 3$, the definition of complete k-consistency for VCSPs given in this paper leads to a generalisation of path consistency in CSPs to the valued constraint framework. An alternative generalisation of path consistency, called 3-cyclic consistency, was presented in a previous paper [7]. The essential difference is that 3-cyclic consistency does not allow the creation of order-3 constraints. This means that, on a

3-variable problem, 3-cyclic consistency is weaker than complete 3-consistency. However, when applying complete 3-consistency, weights that are shifted up to an order-3 constraint ϕ_{ijk} are no longer available for use on other size-3 subproblems; this simply cannot occur in 3-cyclic consistency. Thus these two generalisations of path consistency should be considered as complementary. If we apply both 3-cyclic consistency and complete 3-consistency operations, it is preferable to apply 3-cyclic consistency operations first, so that weights are shifted up to order-3 constraints only as a last resort. Note that 3-cyclic consistency can be established in $O(d^3n^4)$ time on an instance of MAX-CSP [7].

In this paper we have given a polynomial-time algorithm to establish complete k-consistency in VCSPs, where k is any constant. Polynomial-time algorithms exist to simultaneously establish arc consistency and directional arc consistency [6, 17]. It is an open theoretical question whether a form of directional k-consistency can be established in polynomial time for $k > 2$.

The SIP (Semi-Independent Partitioning) approach of Larkin [15] (a strict generalisation of the mini-bucket approach of Dechter [10, 11]) produces a lower bound for a VCSP by completely solving a sequence of subproblems C_1, $C_2, \ldots$ each of whose constraint graphs has induced width $w^* \leq k$ for some constant k. For $i = 1, 2, \ldots$, the constraints in the subproblem C_i are replaced by a single order-k valued constraint $\phi_{P_i} = \text{PROJ}_{P_i} Val_{C_i}$, representing the solutions to C_i projected onto some set P_i of k variables. One can view the calculation of $\phi_{P_i} = \text{PROJ}_{P_i} Val_{C_i}$ (and hence the whole SIP algorithm) as the application of a sequence of variable eliminations and each variable elimination as a k-consistency operation. If this same sequence of k-consistency operations were applied to the original problem, the lower bound could, in fact, be improved since no constraints are actually eliminated (as they are in SIP). Furthermore, since consistency operations are equivalence-preserving transformations, it is clear that many different SIPs could be applied (without actually eliminating constraints) to try to improve the lower bound. We make these remarks simply to illustrate that different heuristics to determine a lower bound can be improved by a greater theoretical understanding of high-order consistency operations in VCSPs.

Face consistency can be used in a similar way when the constraint graph of a VCSP V is not planar. If V_1, V_2, ... are subproblems of V such that the constraint graph of each V_i is planar and degree-bounded, then face consistency can be applied successively to each V_i [8]. Full directional arc consistency [6], 3-cyclic consistency [7], face consistency and a SIP-like k-consistency approach (as described above) are complementary consistency methods for finding a lower bound, and hence could all be applied to the same problem.

5. Conclusion

We have shown that we only need to consider consistency algorithms over two types of discrete valuation structures. A generic definition of high-order consistency has been given together with a polynomial-time algorithm to enforce it for these two types of

valuation structures. This generic definition of consistency in VCSPs provides the possibility of tailoring consistency to particular applications.

One specific version of consistency, called face consistency, has been defined for which an efficient enforcing algorithm exists [8]. Face consistency, applied to subproblems whose constraint graph is planar, provides a level of consistency which may be applied when the cost of establishing complete k-consistency proves to be prohibitive.

Appendix A: Characterisation of Fair Valuation Structures

The following result follows directly from results proved in [9]. It implies that idempotent valuations divide a valuation structure into independent slices.

Lemma A.1 *Let $S = \langle E, \oplus, \geq \rangle$ be a fair valuation structure. Let $\beta, \gamma \in E$, $\beta \leq \gamma$ and let $\alpha_0, \alpha_1 \in E$ be idempotent valuations such that $\alpha_0 \leq \gamma \leq \alpha_1$. Then $\alpha_0 \leq (\gamma \oplus \beta) \leq \alpha_1$ and $\alpha_0 \leq (\gamma \ominus \beta) \leq \alpha_1$. Furthermore, if $\beta \leq \alpha_0$ then $\gamma \oplus \beta = \gamma \ominus \beta = \gamma$.*

Definition A.2 The underlying FCSP (fuzzy constraint satisfaction problem) *of a fair VCSP $= \langle N, D, C, S \rangle$ is a VCSP $V_F = \langle N, D, C', S' \rangle$* in which each constraint function ϕ_P of V has been replaced by ϕ'_P where

$$\forall t \in L(P), \phi'_P(t) = \phi_P(t) \ominus \phi_P(t)$$

over the valuation structure $S' = \langle \{ \phi_P(t) \ominus \phi_P(t) : P \in C \wedge t \in L(P) \} \cup \{\bot, \top\}, \text{MAX}, \geq \rangle$.

By Lemma 1.18, $\alpha \ominus \alpha$ is a maximal idempotent lower bound for α. If α and β share the same maximal idempotent lower bound (i.e., $\alpha \ominus \alpha = \beta \ominus \beta$), then we say that they belong to the same slice. As we will show below, solving the underlying FCSP of a VCSP V tells us in which slice to look to solve V.

Notation If $\alpha \in E$ is idempotent, then the interior of its slice I_α is $\{\gamma \in E : (\gamma \ominus \gamma) = \alpha < \gamma\}$, the set of non-idempotent valuations whose maximal idempotent lower bound is α.

Definition A.3. In a fair valuation structure $S = \langle E, \oplus, \geq \rangle$, if $\alpha \in E$ is idempotent then the α-clipping function $\psi_\alpha : E \to E$ is given by

$$\psi_\alpha(\beta) = \begin{cases} \top & \text{if } \beta \ominus \beta > \alpha \\ \bot & \text{if } \beta \leq \alpha \\ \beta & \text{otherwise} \end{cases}$$

for all $\beta \in E$.

The α-clipping function preserves valuations in I_α, transforms valuations above I_α to $\top$ and those below I_α to $\bot$.

Definition A.4. The slice VCSP of a fair VCSP $V = \langle N, D, C, \langle E, \oplus, \geq\rangle\rangle$ at level $\alpha \in E$ (where α is idempotent) is the VCSP $V_\alpha = \langle N, D, C_\alpha, S_\alpha\rangle$ in which each constraint function ϕ_P of V has been replaced in V_α by $\phi'_P = \phi_P \circ \psi_\alpha$, i.e.,

$$\phi'_P(t) = \begin{cases} \top & \text{if } \phi_P(t) \ominus \phi_P(t) > \alpha \\ \bot & \text{if } \phi_P(t) \leq \alpha \\ \phi_P(t) & \text{otherwise} \end{cases}$$

over the valuation structure $S_\alpha = \langle E_\alpha, \oplus_\alpha, \geq\rangle$, where $E_\alpha = I_\alpha \cup \{\bot, \top\}$ and $\oplus_\alpha$ is given by $\forall \beta, \gamma \in E_\alpha$

$$\beta \oplus_\alpha \gamma = \begin{cases} \top & \text{if } (\beta \oplus \gamma) \ominus (\beta \oplus \gamma) > \alpha \\ \beta \oplus \gamma & \text{otherwise} \end{cases}$$

The following lemma is essential to show that a VCSP can be decomposed into its underlying fuzzy CSP and its slice VCSPs.

Lemma A.5 *In a fair valuation structure $S = \langle E, \oplus, \geq\rangle$, for all α, β, $\gamma \in E$,*

$$\psi_\alpha(\beta \oplus \gamma) = \psi_\alpha(\beta) \oplus_\alpha \psi_\alpha(\gamma)$$

Proof: By symmetry, we only need consider three cases: (a) $\psi_\alpha(\beta) = \psi_\alpha(\gamma) = \bot$, (b) $\psi_\alpha(\beta) = \top$, (c) $\psi_\alpha(\beta) = \beta \wedge \psi_\alpha(\gamma) < \top$.

Case (a) In this case, both β and γ are less than or equal to α which is idempotent. Hence, by Lemma A.1, $\beta \oplus \gamma \leq \alpha$ and $\psi_\alpha(\beta \oplus \gamma) = \bot = \bot \oplus_\alpha \bot = \psi_\alpha(\beta) \oplus_\alpha \psi_\alpha(\gamma)$.

Case (b) It follows from Lemma 1.18(a) that $(\beta \oplus \gamma) \ominus (\beta \oplus \gamma) \geq \beta \ominus \beta$. Thus, since $\psi_\alpha(\beta) = \top$, $(\beta \oplus \gamma) \ominus (\beta \oplus \gamma) \geq \beta \ominus \beta > \alpha$ and hence $\psi_\alpha(\beta \oplus \gamma) = \top = \top \oplus_\alpha \psi_\alpha(\gamma) = \psi_\alpha(\beta) \oplus_\alpha \psi_\alpha(\gamma)$.

Case (c) Since $\psi_\alpha(\beta) = \beta$, we have $\beta > \alpha$ and hence $\beta \oplus \gamma > \alpha$. Thus

$$\psi_\alpha(\beta \oplus \gamma) = \begin{cases} \top & \text{if } (\beta \oplus \gamma) \ominus (\beta \oplus \gamma) > \alpha \\ \beta \oplus \gamma & \text{otherwise} \end{cases}$$

If $\psi_\alpha(\gamma) = \gamma$ then, by definition of $\oplus_\alpha$,

$$\psi_\alpha(\beta) \oplus_\alpha \psi_\alpha(\gamma) = \begin{cases} \top & \text{if } (\beta \oplus \gamma) \ominus (\beta \oplus \gamma) > \alpha \\ \beta \oplus \gamma & \text{otherwise} \end{cases}$$

and we are done. If, on the other hand, $\psi_\alpha(\gamma) = \bot$, then $\gamma \leq \alpha$ and by Lemma A.1, $\beta \oplus \gamma = \beta$. Hence, knowing that $\beta \ominus \beta \leq \alpha$ since $\psi_\alpha(\beta) = \beta$, we can deduce that $\psi_\alpha(\beta \oplus \gamma) = \beta = \beta \oplus_\alpha \bot = \psi_\alpha(\beta) \oplus_\alpha \psi_\alpha(\gamma)$. ■

The following result follows immediately from repeated applications of Lemma A.5.

Lemma A.6 *In a fair VCSP* $\langle N, D, C, \langle E, \oplus, \geq\rangle\rangle$, $\forall \alpha \in E$, $\forall t \in L(N)$,

$$Val_{V_\alpha}(t) = \psi_\alpha(Val_V(t))$$

Theorem A.7 *Any fair VCSP V can be solved by (1) solving the underlying FCSP, and (2) solving a single slice VCSP.*

Proof: Let y be an optimal solution to the underlying FCSP V_F and let $Val_{V_F}(y) = \alpha$ (which is necessarily idempotent from the definition of V_F). Then let t be an optimal solution to the slice VCSP V_α and let $Val_{V_\alpha}(t) = \chi$. We will show that (1) if $\chi < \top$ then t is an optimal solution to V, and that (2) if $\chi = \top$ then y is an optimal solution to V.

(1) From the definition of the underflying FCSP V_F, $\forall z \in L(N)$, $Val_{V_F}(z) = \text{MAX}\{\phi_P(\pi_P z) \ominus \phi_P(\pi_P z) : P \in C\} \leq \oplus_{P \in C} \phi_P(\pi_P z) = Val_V(z)$. From the optimality of y, $\forall z \in L(N)$, $\alpha = Val_{V_F}(y) \leq Val_{V_F}(z) \leq Val_V(z)$. Therefore, if $Val_V(t) = \alpha$ then clearly t is an optimal solution to V. Thus it only remains to consider the case $Val_V(t) > \alpha$.

Since, by Lemma A.6, $\psi_\alpha(Val_V(t)) = Val_{V_\alpha}(t) = \chi < \top$, we can deduce from the definition of ψ_α that $\psi_\alpha(Val_V(t)) = Val_V(t)$. Since t is an optimal solution to V_α, $\forall z \in L(N)$, $Val_{V_\alpha}(z) \geq Val_{V_\alpha}(t)$, and hence, by Lemma A.6, $\psi_\alpha(Val_V(z)) \geq \psi_\alpha(Val_V(t)) = (Val_V(t)) = Val_V(t) > \alpha$. By the definition of ψ_α, either $\psi_\alpha(Val_V(z)) = Val_V(z) \geq Val_V(t)$ or $Val_V(z) \ominus Val_V(z) > \alpha$. In the latter case, knowing that $Val_V(t) \ominus Val_V(t) = \alpha$ (since $\psi_\alpha(Val_V(t)) < \top$), we can deduce from Lemma 1.18(a) that $Val_V(z) > Val_V(t)$. Thus, in all cases $\forall z \in L(N)$, $Val_V(z) \geq Val_V(t)$ and hence t is an optimal solution to V.

(2) By the optimality of t as a solution to V_α, we must have $\forall z \in L(N)$, $Val_{V_\alpha}(z) \geq Val_{V_\alpha}(t) = \chi = \top$. This implies, by Lemma A.6, that $\psi_\alpha(Val_V(z)) = \top$, i.e., that $\alpha_z > \alpha$, where $\alpha_z = Val_V(z) \ominus Val_V(z)$. Then, by the definition of the underlying FCSP V_F, $Val_{V_F}(y) = \text{MAX}\{\phi_P(\pi_P y) \ominus \phi_P(\pi_P y) : P \in C\} = \alpha < \alpha_z$. We can deduce from Lemma 1.18(a) that $\forall P \in C$, $\phi_P(\pi_P y) < \alpha_z$, since α_z is idempotent. By Lemma A.1, $Val_V(y) = \oplus_{P \in C} \phi_P(\pi_P y) \leq \alpha_z = Val_V(z) \ominus Val_V(z)$. Thus, by monotonicity, $\forall z \in L(N)$, $Val_V(y) \leq Val_V(z)$, and hence y is an optimal solution to V. ■

Note that a slice VCSP has the property that the only idempotent valuations in its valuation structure are $\bot$ and $\top$.

Theorem A.8 *If a valuation structure* $S = \langle E, \oplus, \geq\rangle$ *is discrete, fair and has only two idempotent valuations* $\bot$, $\top$, *then it is isomorphic to either* $\mathbf{N} \cup \{\infty\}$ *or* S_m *for some* $m \in \mathbf{N}$.

Proof: If $E = \{\bot, \top\}$, then S is isomorphic to S_1. So suppose $\exists \gamma \in E$ such that $\top > \gamma > \bot$. Let $\alpha_1 = \text{MIN}\{\alpha \in E : \gamma \geq \alpha > \bot\}$ (which exists by discreteness). Define the valuations α_i recursively as follows: $\alpha_0 = \bot$, $\alpha_i = \alpha_{i-1} \oplus \alpha_1$ (for $i \geq 2$). Now, suppose there exists $\beta \in E$ such that $\forall i \in \mathbf{N}$, $\beta \neq \alpha_i$ and $\beta \neq \top$.

If $\alpha_i < \beta < \alpha_{i+1}$ then $\bot < \beta \ominus \alpha_i < \alpha_1$ since $\alpha_i \oplus \bot = \alpha_i < \beta$ and $\alpha_i \oplus \alpha_1 = \alpha_{i+1} > \beta$. But this contradicts the definition of α_1.

If, on the other hand, $\forall i \in \mathbf{N}$, $\beta > \alpha_i$, then this contradicts the discreteness of the valuation structure, unless there are only a finite number of distinct α_i. If $\alpha_m = \alpha_p$, for some $p > m \geq 1$, then $\alpha_m \leq \alpha_{m+1} \leq \alpha_p = \alpha_m$ and hence $\alpha_{m+1} = \alpha_m$. It follows by an easy inductive argument that $\forall i \geq 0$, $\alpha_{m+i} = \alpha_m$. Thus $\alpha_m \oplus \alpha_m = \alpha_{2m} = \alpha_m$ and hence α_m is idempotent. Since $\alpha_m \geq \alpha_1 > \bot$, by the hypothesis of the theorem, we must have $\alpha_m = \top$. But then $\beta > \alpha_m$ is a contradiction.

We can therefore deduce that the valuation structure is the set $\{\alpha_i : i \in \mathbf{N}\} \cup \{\top\}$ which is isomorphic to $\mathbf{N} \cup \{\infty\}$ or S_m (where m is the smallest integer for which α_m is idempotent). ■

References

1. Bertele, U., & Brioschi, F. (1972). *Nonserial Dynamic Programming*. Academic Press, New York, NY, USA.
2. Bistarelli, S., Fargier, H., Montanari, U., Rossi, F., Schiex, T., & Verfaillie, G. (1999). Semiring-based CSPs and valued CSPs: Frameworks, properties and comparison. *Constraints* 4: 199–240.
3. Cooper, M. C. (1990). An optimal *k*-consistency algorithm. *Artif. Intell.* 41: 89–95.
4. Cooper, M. C. (1993). Interpretation of line drawings of complex objects. *Image Vis. Comput.* 11(2): 82–90.
5. Cooper, M. C. (1999). Linear-time algorithms for testing the realisability of line drawings of curved objects. *Artif. Intell.* 108: 31–67.
6. Cooper, M. C. (2003). Reduction operations in fuzzy and valued constraint satisfaction. *Fuzzy Sets Syst.* 134: 311–342.
7. Cooper, M. C. (2004). Cyclic consistency: A local reduction operation for binary valued constraints. *Artif. Intell.* 155(1–2): 69–92.
8. Cooper M. C. (2005). *High-Order Consistency in Valued Constraint Satisfaction*. Internal Report, IRIT, Université Toulouse III.
9. Cooper, M. C., & Schiex, T. (2004). Arc consistency for soft constraints. *Artif. Intell.* 154(1–2): 199–227.
10. Dechter, R. (1997). Mini-buckets: A general scheme for generating approxiamtions in automated reasoning. In *Proc. IJCAI-97*, Nagoya, Japan, pages 1297–1303.
11. Dechter, R. (2003). *Constraint Processing*. San Mateo, CA: Morgan Kaukmann.
12. Dechter, R., & Pearl, J. (1988). Network-based heuristics for constraint satisfaction problems. *Artif. Intell.* 34: 1–38.
13. Fargier, H., & Lang, J. (1993). Uncertainty in constraint satisfaction problems: A probabilistic approach. In *Proc. ECSQARU*, Springer-Verlag, LNCS 747, pages 97–104.
14. Fargier, H., Lang, J., & Schiex, T. (1993). Selecting preferred solutions in Fuzzy Constraint Satisfaction Problems. In *Proc. of the 1st European Congress on Fuzzy and Intelligent Technologies*.
15. Larkin, D. (2003). Semi-independent partitioning: A method for bounding the solution to COP's. In *Proc. Principles and Practice of Constraint Propgramming—CP 2003*, Springer-Verlag, LNCS 2833, pages 894–898.
16. Larrosa, J. (2002). On arc and node consistency in weighted CSP. In *Proc. AAAI'02*.
17. Larrosa, J., & Schiex, T. (2003). *In the Quest of the Best Form of Local Consistency for Weighted CSP*. IJCAI.
18. Rosenfeld, A., Hummel, R., & Zucker, S. (1976). Scene labelling by relaxation operations. *IEEE Trans. Syst. Man Cybern.* 6(6): 173–184.
19. Schiex, T. (2000). Arc consistency for soft constraints. In *Proc. CP'2000, LNCS 1894*, pages 411–424.
20. Schiex, T., Fargier, H., & Verfaillie, G. (1995). Valued constraint satisfaction problems: Hard and easy problems. In *Proc. of the 14th IJCAI*, Montreal, Canada, pages 631–637.