Available online at www.sciencedirect.com
SCIENCE DIRECT
Artificial Intelligence 155 (2004) 69–92
Artificial Intelligence
www.elsevier.com/locate/artint

Cyclic consistency: A local reduction operation for binary valued constraints

Martin C. Cooper

IRIT, University of Toulouse III, 31062 Toulouse, France

Received 14 January 2002; received in revised form 28 March 2003

Abstract

Valued constraint satisfaction provides a general framework for optimisation problems over finite domains. It is a generalisation of crisp constraint satisfaction allowing the user to express preferences between solutions.

Consistency is undoubtedly the most important tool for solving crisp constraints. It is not only a family of simplification operations on problem instances; it also lies at the heart of intelligent search techniques [G. Kondrak, P. van Beek, Artificial Intelligence 89 (1997) 365–387] and provides the key to solving certain classes of tractable constraints [P.G. Jeavons, D.A. Cohen, M.C. Cooper, Artificial Intelligence 101 (1998) 251–265].

Arc consistency was generalised to valued constraints by sacrificing the uniqueness of the arc consistency closure [M.C. Cooper, T. Schiex, Artificial Intelligence, in press]. The notion of 3-cyclic consistency, introduced in this paper, again sacrifices the unique-closure property in order to obtain a generalisation of path consistency to valued constraints which is checkable in polynomial time. In MAX-CSP, 3-cyclic consistency can be established in polynomial time and even guarantees a local form of optimality. The space complexity of 3-cyclic consistency is optimal since it creates no new constraints.

Keywords: Valued constraint satisfaction problem (VCSP); Constraint propagation; Arc consistency; Path consistency; In-scope reductions; Local irreducibility; MAX-CSP

1. Introduction

The constraint satisfaction problem (CSP) is a generic combinatorial problem over finite domains. Most of the techniques developed for solving CSPs make use of the concept of

E-mail address: cooper@irit.fr (M.C. Cooper).

0004-3702/$ – see front matter
doi:10.1016/j.artint.2003.06.004

local consistency: if a legal labelling x for some set of variables U cannot be extended to a legal labelling for $V \supset U$, then x cannot be extended to a legal global labelling and can hence be eliminated. This is called an order-k consistency operation if $|V| = k$. It is known as arc consistency if $|V| = 2$ and $|U| = 1$ and path consistency if $|V| = 3$ and $|U| = 2$. Optimal algorithms have been developed for arc [15,18], path [13] and k-consistency [6]. Consistency operations can be used as a preprocessing step to simplify a CSP or during exhaustive search to prune the search tree.

Arc consistency has been successfully extended to the Valued Constraint Satisfaction Problem (VCSP) [8,9,16]. In the VCSP, the aim is to find an assignment that minimises the aggregate of constraint violations, and thus extends the CSP to include a wide range of optimisation problems [2,17]. The establishment of arc consistency transforms a CSP into a unique equivalent problem in polynomial time. In VCSPs, to ensure equivalence and efficiency, the uniqueness of the arc consistency closure has to be sacrificed (except in the very special case of an idempotent aggregation operator [2]). Establishing arc consistency in a VCSP involves projecting penalties from binary constraints to domains, which must then be compensated for by decreasing the weights on the binary constraint. Thus a necessary condition for arc consistency to be applied to a VCSP is that the aggregation operator possesses an inverse. This condition is satisfied by all commonly-employed optimising versions of the CSP, notably MAX-CSP [1,12, 14].

Consistency operations in VCSPs facilitate the search for an optimal solution by providing a tighter lower bound on the valuation of all solutions. The notion of 3-cyclic consistency introduced in this paper corresponds to a state in which no set of arc consistency operations applied simultaneously to a 3-variable subproblem of the VCSP can increase this lower bound.

The use of path consistency in CSPs has been limited by its space complexity. Indeed, in the worst case, establishing path consistency involves creating constraints between all pairs of variables, however unrelated they were in the original problem [4,5]. This has led to the definition of certain restricted forms of path consistency which do not create new constraints [10]. In VCSPs, the problem of space complexity is aggravated even further. The projection of penalties from a set V of cardinality 3 onto a subset U, and its consequent compensation within V, would require the creation of an order 3 constraint on V. We will show that 3-cyclic consistency, on the other hand, does not require the creation of any new binary or ternary constraints.

2. VCSP: notation and definitions

Valued CSPs (or VCSPs) were initially introduced in [17]. An alternative formulation of soft constraint satisfaction was given independently in [3] based on semirings. One can consider VCSPs as the very important special case of semiring based CSPs in which the set of valuations (penalties) possesses a total order, which not only covers the most important applications but also allows us to use a simpler notation.

A valued constraint satisfaction problem (VCSP) is composed of a set of n variables N, a set D of variable domains, a set $\mathcal{C}$ of constraints and a valuation structure S. Each constraint $C(P) \in \mathcal{C}$ is a pair (P, ϕ_P), where $P \subseteq N$ is the constraint scope and ϕ_P is the constraint function associating a penalty to each assignment x of values to the variables in P, i.e., $\phi_P(x)$ is the degree to which x violates the constraint $C(P)$. The projection of a tuple of values t onto a set of variables $P \subseteq N$ is denoted by $\Pi_P t$. Finally, the valuation of a solution t, a tuple of n values, is the aggregate of the penalties $\phi_P(\Pi_P t)$ over all constraints $C(P) \in \mathcal{C}$.

The valuation structure S is a triple $\langle E, \oplus, \geqslant \rangle$ composed of the set of possible valuations E, the operator (denoted by $\oplus$) used for aggregating penalties and the total order $\geqslant$ used to compare valuations of different tuples. The maximum element $\top$ of E represents total inconsistency, whereas its minimum element $\bot$ represents total consistency. The aggregation operator must satisfy a set of properties that are captured by a set of axioms defining a valuation structure.

Definition 2.1. A *valuation structure* is defined as a tuple $\langle E, \oplus, \geqslant \rangle$ such that:

- E is a set, whose elements are called valuations, which is totally ordered by $\geqslant$, 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:
  - Identity:
  $$\forall \alpha \in E, \quad (\alpha \oplus \bot) = \alpha;$$
  - Monotonicity:
  $$\forall \alpha, \beta, \gamma \in E, \quad (\alpha \geqslant \beta) \Rightarrow (\alpha \oplus \gamma) \geqslant (\beta \oplus \gamma);$$
  - Absorbing element:
  $$\forall \alpha \in E, \quad (\alpha \oplus \top) = \top.$$

The valuation structure is known as *strictly monotonic* if it also satisfies the following axiom:

- Strict monotonicity:
$$\forall \alpha, \beta, \gamma \in E, \quad (\alpha > \beta) \wedge (\gamma \neq \top) \Rightarrow (\alpha \oplus \gamma) > (\beta \oplus \gamma).$$

For a more detailed analysis and justification of these axioms, we invite the reader to consult [2,17]. MAX-CSP, the problem of maximising the number of satisfied constraints in a constraint satisfaction problem (CSP), can be expressed as a VCSP over the valuation structure $\langle \mathbb{N} \cup \{\infty\}, +, \geqslant \rangle$, although the valuation ∞ is never attained.

Definition 2.2. A *valued CSP* (*VCSP*) is a tuple $\langle N, D, \mathcal{C}, S \rangle$ where N is a set of n variables $N = \{1, \ldots, n\}$, each variable $i \in N$ has a finite domain of possible values A_i, $D = \{A_1, \ldots, A_n\}$, $\mathcal{C}$ is a set of constraints, and $S = \langle E, \oplus, \geqslant \rangle$ is a valuation structure. Each constraint $C(P) = (P, \phi_P)$ in $\mathcal{C}$ is composed of a set of variables (its scope) $P \subseteq N$ and a function ϕ_P from the Cartesian product of the domains A_i $(i \in P)$ to E.

Definition 2.3. A *binary VCSP* is a VCSP in which the arity $|P|$ of each constraint $C(P) \in \mathcal{C}$ is no greater than 2.

Notation. A VCSP(sm), or strictly monotonic VCSP, is a VCSP whose valuation structure is strictly monotonic.

An assignment t of values to some variables $V \subseteq N$ can be evaluated by simply aggregating, for all assigned constraints $C(P)$ (i.e., constraints such that $P \subseteq V$), the valuations of the tuples $\Pi_P t$.

Definition 2.4. In a VCSP $\mathcal{P} = \langle N, D, \mathcal{C}, S\rangle$, the *valuation* of an assignment t to a set of variables $V \subseteq N$ is defined by:

$$\mathrm{Val}_{\mathcal{P}}(t) = \bigoplus_{C(P)\in\mathcal{C}\wedge P\subseteq V} \big(\phi_P(\Pi_P t)\big).$$

The problem usually considered is to find a complete assignment $t \in A_1 \times \cdots \times A_n$ with a minimum valuation.

Notation. If $P \subseteq N$, then $L(P)$ represents the set of possible labellings for P, i.e. the cartesian product of the domains A_i for $i \in P$.

Definition 2.5. Two VCSPs $V_1 = \langle N, D, \mathcal{C}_1, S\rangle$ and $V_2 = \langle N, D, \mathcal{C}_2, S\rangle$ are *equivalent* if all tuples $x \in L(N)$ have identical valuations in V_1 and V_2.

Definition 2.6. The *subproblem* of a VCSP $V = \langle N, D, \mathcal{C}, S\rangle$ on $J \subseteq N$ is the VCSP $V_J = \langle J, D_J, \mathcal{C}_J, S\rangle$, where $D_J = \{A_j\colon\ j \in J\}$ and $\mathcal{C}_J = \{C(P) \in \mathcal{C}\colon\ P \subseteq J\}$.

Definition 2.7. For a VCSP V, an *equivalence-preserving transformation* of V on $J \subseteq N$ is an operation which transforms the subproblem of V on J into an equivalent VCSP.

Arc consistency operations [9,16] are an example of an equivalence-preserving transformation. To establish arc consistency in VCSPs, we have to shift weights from one constraint to another; to do this we have to be able to compensate for the addition of α in one constraint by the subtraction of α from another. This is made possible by the following additional axiom:

Definition 2.8 (*from* [9]). In a valuation structure $S = \langle E, \oplus, \geqslant\rangle$, if $u, v \in E$, $u \geqslant v$, and there exists a valuation $w \in E$ such that $w \oplus v = u$, then w is known as a difference of u and v. The valuation structure S is *fair* if for any pair of valuations $u, v \in E$, with $u \geqslant v$, there exists a maximal difference of u and v. This unique maximal difference of u and v is denoted by $u \ominus v$.

This simple axiom is actually satisfied by most existing concrete soft constraint frameworks, including all those with a strictly monotonic operator $\oplus$ (see [8] for a formal proof of this result). In this article we restrict our attention to strictly monotonic valuation structures. The following theorem will allow us to greatly simplify the notation in the rest of the article.

Theorem 2.9. *Let $S = \langle E, \oplus, \geqslant \rangle$ be a strictly monotonic valuation structure. Then the set of non-$\top$ valuations in S can be embedded in a totally-ordered strictly monotonic additive abelian group.*

Proof. It is known that any strictly monotonic valuation structure can be embedded in a fair valuation structure [8]. Thus we can assume that S is a fair valuation structure with difference operator $\ominus$. Let $E' = (E - \{\top\}) \cup \{-\alpha\colon \alpha \in E - \{\bot, \top\}\}$. The operator $\oplus$ is extended to E' as follows (and renamed $+$ in the process, to comply with the standard notation for additive groups):

$$\forall \alpha, \beta \in E - \{\top\}, \quad \alpha + \beta = \alpha \oplus \beta;$$
$$\forall \alpha, \beta \in E - \{\bot, \top\}, \quad -\alpha + -\beta = -(\alpha \oplus \beta);$$
$$\forall \alpha \in E - \{\bot, \top\},\ \forall \beta \in E - \{\top\}, \quad (\alpha \leqslant \beta) \Rightarrow -\alpha + \beta = \beta \ominus \alpha;$$
$$\forall \alpha \in E - \{\bot, \top\},\ \forall \beta \in E - \{\top\}, \quad (\beta < \alpha) \Rightarrow -\alpha + \beta = -(\alpha \ominus \beta).$$

The order $\leqslant$ is extended to E' as follows: $\forall \alpha \in E - \{\bot, \top\},\ \forall \beta \in E - \{\top\},\ -\alpha < \beta$; $\forall \alpha, \beta \in E - \{\bot, \top\},\ -\alpha < -\beta$ iff $\beta < \alpha$. It is easily verified that E' is an abelian group, with identity element $\bot$, satisfying the strict monotonicity property:

$$\forall \alpha, \beta, \gamma \in E', \quad (\alpha > \beta) \Rightarrow \alpha \oplus \gamma > \beta \oplus \gamma.$$

Since E' is a group, the operator $+$ has an inverse which we denote by $-$. We can, in fact, extend both $+$ and $-$ to $E' \cup \{\top\} \times E'$ in the obvious way: $\top + \alpha = \top$ and $\top - \alpha = \top$. The total ordering $\leqslant$ also has an obvious extension to $E' \cup \{\top\}$. Note, however, that $\alpha - \top$ is undefined for all $\alpha \in E' \cup \{\top\}$. □

3. Arc consistency operations in VCSPs

In this section we review arc consistency operations in VCSPs and demonstrate the existence of order-3 consistency operations in VCSPs which are stronger than arc consistency.

Definition 3.1. The *underlying CSP* of a VCSP V has the same variables as V together with, for each constraint $C(P) = (P, \phi_P)$ in $\mathcal{C}$, a crisp constraint $C'(P)$ satisfying $\forall t \in L(P)$ $(t \in C'(P) \Leftrightarrow \phi_P(t) < \top)$ (i.e., $t \in C'(P)$ iff t is not a totally forbidden labelling).

Definition 3.2 (*from* [9]). A binary VCSP(sm) is *arc consistent* if

(1) its underlying CSP is arc consistent;
(2) $\forall i, j \in N$ such that i constraints j, $\forall a \in A_i$, if $\phi_i(a) < \top$ then $\exists b \in A_j$ such that $\phi_{ij}(a, b) = \bot$.

Condition (2) of Definition 3.2 says that non-$\top$ penalties are projected as much as possible from binary to unary constraints. Consider the 3-variable VCSP V shown in Fig. 1(a). It is an instance of MAX-SAT, a problem which consists in satisfying the maximum number of crisp constraints on Boolean variables. This instance V comprises the three constraints $X_1 \vee X_2$, $\neg X_1 \wedge X_3$, $\neg X_2 \wedge X_3$. Thus, for example, $\phi_{12}(X_1, X_2) = 1$ if $X_1 = X_2 =$ false and $\phi_{12}(X_1, X_2) = 0$ otherwise. Each line in Fig. 1(a) joining value a for X_i with value b for X_j represents a penalty $\phi_{ij}(a, b) = 1$. Fig. 1(b) shows an arc consistency closure of V (obtained by projecting penalties from binary to unary constraints). For example, the penalties $\phi_{13}(T, T) = \phi_{13}(T, F) = 1$ in Fig. 1(a) have been replaced by the penalty $\phi_1(T) = 1$ in Fig. 1(b). However, the arc consistency closure is not unique; Fig. 1(c) shows a different arc consistency closure of V.

Definition 3.3 (*from* [9]). A binary VCSP(sm) is *directional arc consistent* if $\forall i, j \in N$ such that i constraints j and $i < j$, $\forall a \in A_i$, if $\phi_i(a) < \top$ then $\exists b \in A_j$ such that $\phi_{ij}(a, b) = \phi_j(b) = \perp$.

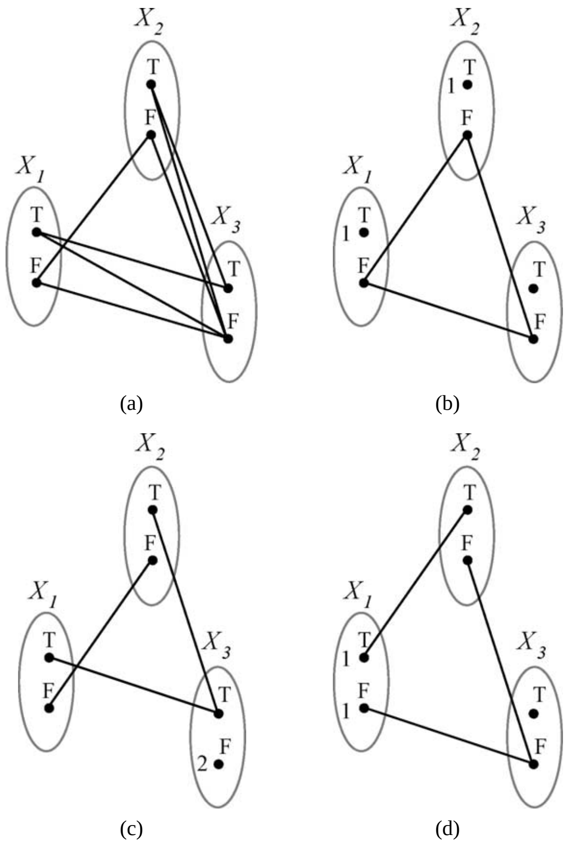

Fig. 1. (a) A VCSP V; (b), (c) two arc consistency closures of V; (d) a full directional arc consistency closure of V.

Definition 3.4. A binary VCSP(sm) is *full directional arc consistent* if it is both arc consistent and directional arc consistent.

Fig. 1(d) shows a full directional arc consistency closure of V (obtained by shifting penalties towards earlier variables in the order X_1, X_2, X_3 as well as projecting penalties from binary to unary constraints). Although the four VCSPs shown in Fig. 1 are all equivalent, the version of Fig. 1(d) has the distinct advantage that, since every value for X_1 has a penalty of 1, it is clear that all solutions to V have a penalty of at least 1.

Notation. Let $M(\phi_P) = \mathrm{MIN}\{\phi_P(x)\colon\ x \in L(P)\}$ denote the minimum valuation attained by the constraint function ϕ_P.

Definition 3.5. The function f_{MIN} is given by

$$f_{\mathrm{MIN}}(V) = \bigoplus_{C(P)\in\mathcal{C}} \bigl\{M(\phi_P)\bigr\}.$$

f_{MIN} is the aggregate of the minimum weights in each constraint. It provides a lower bound on the valuations of all solutions to a VCSP V. Having such a lower bound is particularly important in the context of branch and bound search [1,14].

Full directional arc consistency is not always sufficient to render explicit such a lower bound f_{MIN} even on 3-variable instances of MAX-SAT. Fig. 2(a) shows a VCSP representing the instance of MAX-SAT with constraints $X_1 \wedge \neg X_2,\ X_1 \wedge \neg X_3,\ X_2 \vee X_3$. Fig. 2(b) is the full directional arc consistency closure of this instance (which in this case happens to be unique). Fig. 2(c) is an equivalent VCSP which cannot be obtained by applying full directional arc consistency alone but which can be obtained by the sequence of arc consistency operations shown in Fig. 3.

Taking as a sample of problems all 3-variable instances of MAX-SAT with 3 binary constraints, we found by exhaustive computer search that 1699 of the 4096 problems had a lower bound $f_{\mathrm{MIN}} \geqslant 1$. In 66.45% of these 1699 cases, this was detected by full directional

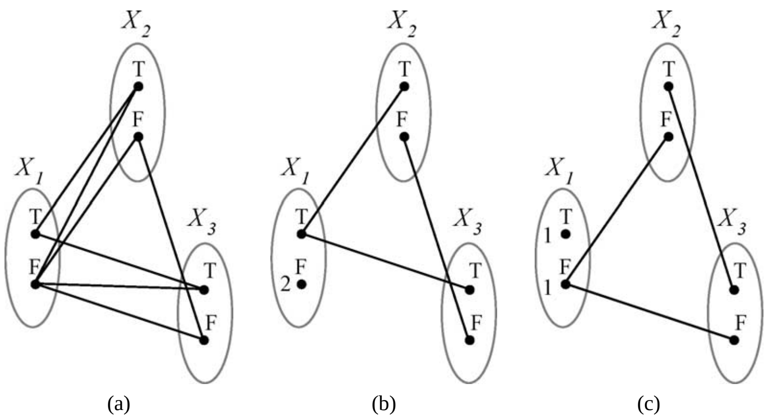

Fig. 2. (a) A VCSP; (b) its full directional arc consistency closure; (c) an equivalent VCSP.

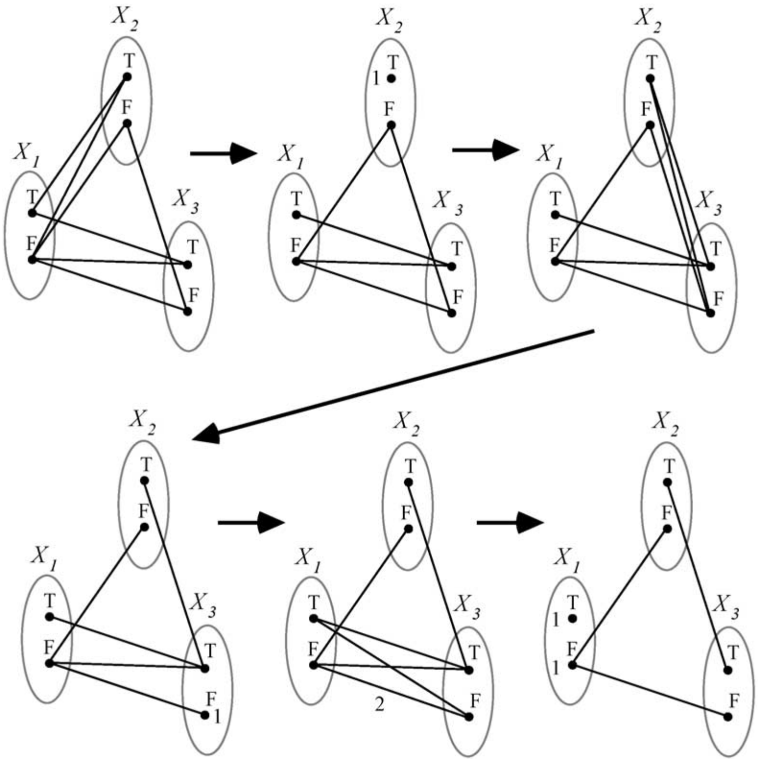

Fig. 3. A sequence of arc consistency operations which transforms the VCSP of Fig. 2(a) into the VCSP of Fig. 2(c).

arc consistency alone, whereas allowing sequences of arc consistency operations, such as illustrated in Fig. 3, increased this percentage to 74.69%.

As the domain size increases, full directional arc consistency alone becomes less effective for determining lower bounds. Fig. 4(a) shows a 3-variable instance of MAX-CSP over size-3 domains. It consists in finding a assignment of values from the domain $\{a, b, c\}$ (where $a < b < c$) to the variables X_1, X_2, X_3 which simultaneously satisfies the greatest number of the following set of six constraints: $X_1 \geqslant c \vee X_2 \geqslant c$; $X_1 \geqslant b \vee X_3 \leqslant a$; $X_1 \leqslant b \vee X_3 \geqslant c$; $X_2 \geqslant b \vee X_3 \geqslant c$; $X_2 \leqslant b \vee X_3 \leqslant a$; $X_3 = b$. Again, each line joining a value u for X_i and a value v for X_j represents a penalty $\phi_{ij}(u, v) = 1$.

Applying a full directional arc consistency algorithm [8] to the VCSP of Fig. 4(a) leaves $f_{\mathrm{MIN}} = 0$, whatever the ordering of the three variables. However, the equivalent problem shown in Fig. 4(b) (for which $f_{\mathrm{MIN}} = 1$) can be obtained by the simultaneous shifting of weights between unary and binary constraints shown in Fig. 4(c). In Fig. 4(c), white arrows represent a shifting of a penalty of 1 from a unary constraint up to a binary constraint, whereas black arrows represent a projection of a penalty of 1 from a binary to a unary constraint. For example, the white arrow leaving value c for X_3 represents the operation:

$$\phi_3(c) := \phi_3(c) - 1; \quad \text{for each } x \in A_1 \text{ do } \phi_{13}(x, c) := \phi_{13}(x, c) + 1$$

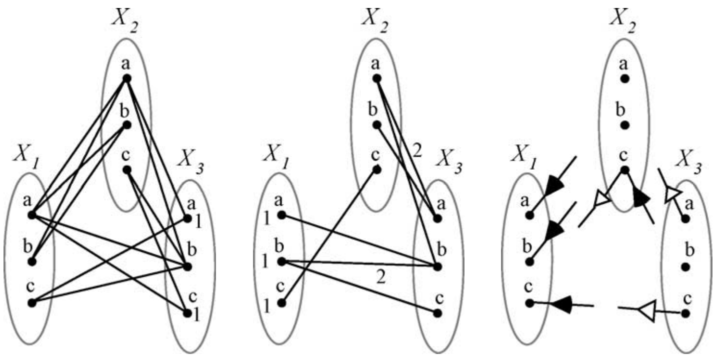

Fig. 4. (a) A VCSP; (b) an equivalent VCSP obtained by the sequence of arc consistency operations shown in (c).

and the black arrow entering value c for X_1 represents the operation:

$$\phi_1(c) := \phi_1(c) + 1; \quad \text{for each } z \in A_3 \text{ do } \phi_{13}(c, z) := \phi_{13}(c, z) - 1.$$

4. Cyclic consistency

The transformations of 3-variable VCSPs given in Figs. 3 and 4(c) are examples of order-3 reduction operations which are stronger than arc consistency or full directional arc consistency alone. The concept of cyclic consistency, defined in this section, when restricted to cycles of length three, provides the key to a generalisation of path consistency from CSPs to VCSPs. A cyclic consistency operation is a set of arc consistency operations applied simultaneously to a cycle of variables $(i_1, i_2, \ldots, i_r)$. If k, i, j are three consecutive variables in the cycle, then $d_i(x)$ is the weight projected down from ϕ_{ij} to $\phi_i(x)$ (i.e., $\forall y \in A_j$ $\phi_{ij}(x, y)$ decreases by $d_i(x)$; $\phi_i(x)$ increases by $d_i(x)$) and $u_i(x)$ is the weight projected up to ϕ_{ki} from $\phi_i(x)$ (i.e., $\phi_i(x)$ decreases by $u_i(x)$; $\forall w \in A_k \phi_{ki}(w, x)$ increases by $u_i(x)$).

Note that, purely for notational convenience, we allow the weights $d_i(x)$ and $u_i(x)$ to be negative. Thus a negative weight $d_i(x) = -\alpha$ shifted from ϕ_{ij} to $\phi_i(x)$, in fact, corresponds to a penalty of α shifted from $\phi_i(x)$ to ϕ_{ij}.

Consider, as an example, the cycle of variables (X_1, X_2, X_3) in the VCSP in Fig. 4. In the set of arc consistency operations of Fig. 4(c), the weights projected from ϕ_{12} onto ϕ_1 are $d_1(a) = 1,\ d_1(b) = 1,\ d_1(c) = 0$ (corresponding to the two arrows entering values a, b for X_1 from the direction of X_2). The weights projected from ϕ_2 up to ϕ_{12} are $u_2(a) = 0,\ u_2(b) = 0,\ u_2(c) = 1$ (corresponding to the arrow leaving value c for X_2 in the direction of X_1). In Fig. 4(c), arrows pointing in an anticlockwise direction correspond to positive values of $d_i(x)$ and $u_i(x)$, whereas arrows pointing in a clockwise direction correspond to negative values. Thus, for example, the weights projected from ϕ_1 up to ϕ_{13} are $u_1(a) = 0,\ u_1(b) = 0,\ u_1(c) = -1$ (corresponding to the arrow entering value c for X_1 in a clockwise direction from X_3).

A set of arc consistency operations applied to a cycle of variables must leave penalties which still lie in E. Although we allow negative changes ($d_i(x)$ and $u_i(x)$) to penalties, the penalties themselves ($\phi_i(x) + d_i(x) - u_i(x)$ and $\phi_{ij}(x, y) + u_j(y) - d_i(x)$) must remain non-negative. This leads naturally to the following definition of a cyclic consistency operation.

Definition 4.1. Let $V = \langle N, D, \mathcal{C}, \langle E, \oplus, \geqslant\rangle\rangle$ be a binary VCSP(sm) and let E' be the natural extension of $E - \{\top\}$ to an additive abelian group, as described in Theorem 2.9. A *cyclic consistency operation* (CCO) on variables $i_1 < i_2 < \cdots < i_r$ consists of valuations $d_i(x), u_i(x) \in E'$, for each $i \in I = \{i_1, i_2, \ldots, i_r\}$ and for each $x \in A_i$, satisfying

$$\forall i \in I,\ \forall x \in A_i, \quad \phi_i(x) + d_i(x) \geqslant u_i(x); \tag{1}$$

$$\forall (i, j) \in \big\{(i_1, i_2), (i_2, i_3), \ldots, (i_r, i_1)\big\},\ \forall x \in A_i,\ \forall y \in A_j,$$
$$\phi_{ij}(x, y) + u_j(y) \geqslant d_i(x). \tag{2}$$

The result of applying this cyclic consistency operation is to transform the constraint functions ϕ_i, for $i \in I$, to ϕ'_i, where

$$\forall x \in A_i \quad \phi'_i(x) = \big(\phi_i(x) + d_i(x)\big) - u_i(x)$$

and the constraint functions ϕ_{ij}, for $(i, j) \in \{(i_1, i_2), (i_2, i_3), \ldots, (i_r, i_1)\}$ to ϕ'_{ij}, where

$$\forall (x, y) \in A_i \times A_j \quad \phi'_{ij}(x, y) = \big(\phi_{ij}(x, y) + u_j(y)\big) - d_i(x).$$

Let $h \in I,\ x_0 \in A_h$. The CCO is known as (h, x_0)-*increasing* if it also satisfies

$$\begin{aligned} &\forall x \in A_h - \{x_0\}, \quad \phi_h(x) = M(\phi_h) \Rightarrow \phi'_h(x) \geqslant M(\phi_h),\\ &\forall x \in A_h - \{x_0\}, \quad \phi_h(x) > M(\phi_h) \Rightarrow \phi'_h(x) > M(\phi_h),\\ &\phi_h(x_0) = M(\phi_h) \wedge \phi'_h(x_0) > M(\phi_h). \end{aligned} \tag{3}$$

The CCO is known as h-*increasing* if it is (h, x_0)-increasing for all $x_0 \in A_h$ such that $\phi_h(x_0) = M(\phi_h)$.

The result of applying an (h, x_0)-increasing CCO is to reduce the number of valuations $\phi_h(x)$ equal to the minimum valuation $M(\phi_h)$, whereas an h-increasing CCO actually increases the minimum valuation, i.e., $M(\phi'_h) > M(\phi_h)$. If $M(\phi_j) = \bot$ for all $j \in I - \{h\}$ and $M(\phi_{ij}) = \bot$ for all $i, j \in I$, then an h-increasing CCO increases $f_{\text{MIN}}(V)$. Figs. 3 and 4 show examples of X_1-increasing CCOs on the cycle of variables (X_1, X_2, X_3).

The importance of (h, x_0)-increasing CCOs will become apparent later, when we show that h-increasing CCOs can be efficiently constructed "brick by brick" as a sequence of (h, x_0)-increasing CCOs. We call $r = |I|$ the order of the operation. We will concentrate on order-3 CCO's.

Theorem 4.2. *A cyclic consistency operation in a* VCSP(sm) *is an equivalence-preserving transformation.*

Proof. By definition of a CCO, the new valuations $\phi'_i(x), \phi'_{ij}(x, y)$ all lie in E. Let V' be the VCSP which results when a CCO is applied to a VCSP V. It is easy to verify that,

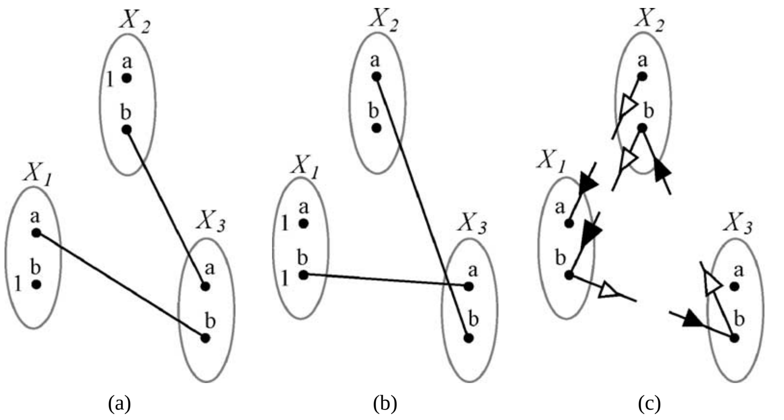

Fig. 5. (a) A VCSP V over the valuation structure $\langle \mathbb{N} \cup \{\infty\}, +, \geqslant \rangle$; (b) an equivalent VCSP V' in which $f_{\mathrm{MIN}} = 1$; (c) a CCO which transforms V into V'.

for all tuples $x \in L(N)$, $\mathrm{Val}_{V'}(x) = \mathrm{Val}_V(x)$, since each $d_i(x)$ or $u_i(x)$ occurs twice in $\mathrm{Val}_{V'}(x)$, once unnegated and once negated. □

Note that it is essential that $d_i(x) < \top$ and $u_i(x) < \top$ for this theorem to be valid.

Definition 4.3. A binary VCSP(sm) V is *cyclic consistent* on variables I, if no h-increasing CCO exists on I, where $h = \min(I)$.

The asymmetry of Definition 4.3 is only an illusion. Indeed we can easily show that, for all $h, i \in I$, if an i-increasing CCO exists on I then an h-increasing CCO exists on I. Let $I = \{i_1, i_2, \ldots, i_r\}$ and let $C(h,i)$ be the set of variables of the cycle $(i_1, \ldots, i_r)$ between $h = i_s$ and $i = i_t$ (i.e., if $s \leqslant t$ then $C(h,i) = \{i_s, \ldots, i_t\}$; if $s > t$ then $C(h,i) = \{i_s, \ldots, i_r, i_1, \ldots, i_t\}$). Suppose that an i-increasing CCO exists (given by the functions d_j, u_j) which increases $M(\phi_i)$ by δ. Define $d'_j(x) = d_j(x) + \delta$ for $j \in C(h,i) - \{i\}$; $d'_j(x) = d_j(x)$ otherwise. Define $u'_j(x) = u_j(x) + \delta$ for $j \in C(h,i) - \{h\}$; $u'_j(x) = u_j(x)$ otherwise. Then the CCO given by the functions d'_j, u'_j is an h-increasing CCO.

Examples of CCOs were given in Figs. 3 and 4. Another example of a CCO is shown in Fig. 5. In this case, even though no constraint exists between variables X_1 and X_2 in the problem V of Fig. 5(a), a CCO can still increase f_{MIN} by transforming V into the equivalent VCSP V' of Fig. 5(b). A CCO which transforms V into V' is shown in Fig. 5(c) (with black arrows representing projections of a weight of 1 from a binary constraint to a unary constraint and white arrows representing a shifting of 1 from a unary to a binary constraint). This CCO is given by $u_1(a) = d_2(a) = d_3(a) = u_3(a) = 0$ and $d_1(a) = d_1(b) = u_1(b) = d_2(b) = u_2(a) = u_2(b) = d_3(b) = u_3(b) = 1$. Note that all of the valuations $u_i(x), d_i(x)$ are non-negative since all arrows point in an anticlockwise direction. Clearly $f_{\mathrm{MIN}}(V') = 1 > 0 = f_{\mathrm{MIN}}(V)$ and this is again a X_1-increasing CCO. Note that this transformation has not introduced any new constraints since variables X_1 and X_2 are still not mutually constraining in V'.

In fact, the following theorem shows that we never need to introduce new constraints when establishing cyclic consistency. This is clearly important for the space efficiency of cyclic consistency.

Theorem 4.4. *Let* $V = \langle N, D, \mathcal{C}, \langle E, \oplus, \geqslant \rangle \rangle$ *be a binary* VCSP(sm), *let* $I = \{i_1, i_2, \ldots, i_r\} \subseteq N$, *where* $i_1 < i_2 < \cdots < i_r$, *and let* $h \in I$. *If there exists an* h*-increasing CCO on* I *in* V, *then there exists an* h*-increasing CCO on* I *in* V *which does not create any new constraints.*

Proof. Consider an h-increasing CCO $\{d_k, u_k\colon k \in I\}$. Suppose that the pair of variables $(i, j) \in \{(i_1, i_2), (i_2, i_3), \ldots, (i_r, i_1)\}$ are not mutually constraining in V, i.e., $\forall (x, y) \in A_i \times A_j \phi_{ij}(x, y) = \bot$.

Let x_1 be such that $d_i(x_1) = \text{MAX}\{d_i(x)\colon x \in A_i\}$. Denote $d_i(x_1)$ by δ. Thus, $\forall x \in A_i$ $d_i(x) \leqslant \delta$. Furthermore, by condition (2) of Definition 4.1 of CCO, we must have $\forall y \in A_j$ $u_j(y) \geqslant d_i(x_1) = \delta$, since $\phi_{ij}(x_1, y) = \bot$. Now, define

$$\forall x \in A_i, \quad D_i(x) = \delta, \quad U_i(x) = u_i(x);$$
$$\forall y \in A_j, \quad D_j(y) = d_j(y), \quad U_j(x) = \delta;$$
$$\forall k \in I - \{i, j\},\ \forall z \in A_k, \quad D_k(z) = d_k(z), \quad U_k(z) = u_k(z).$$

Since $\{d_k, u_k\colon k \in I\}$ is an h-increasing CCO, it is easily seen that $\{D_k, U_k\colon k \in I\}$ is also an h-increasing CCO, because

$$\forall x \in A_i, \quad D_i(x) - U_i(x) = \delta - u_i(x) \geqslant d_i(x) - u_i(x),$$
$$\forall y \in A_j, \quad D_j(y) - U_j(y) = d_j(y) - \delta \geqslant d_j(y) - u_j(y)$$

and

$$\forall (x, y) \in A_i \times A_j \quad U_j(y) - D_i(x) = \bot.$$

Furthermore, no new constraint has been introduced since

$$\forall (x, y) \in A_i \times A_j \quad \phi_{ij}(x, y) + U_j(y) - D_i(x) = \phi_{ij}(x, y) = \bot. \quad \Box$$

Definition 4.5. A binary VCSP(sm) V is 3-*cyclic consistent* if V is cyclic consistent on I for all $I \subseteq N$ such that $|I| = 3$.

Note that a 3-cyclic consistent VCSP(sm) is not necessarily arc consistent nor directional arc consistent. Although a CCO can be thought of as a set of arc consistency operations which are simultaneously applied to a cycle of variables, for complexity reasons we do not blindly apply all possible sets of arc consistency operations, but only those which actually improve the expression of the VCSP in terms of f_{MIN}. In the same way that path consistency in CSPs is almost always applied in conjunction with arc consistency (thus establishing strong 3-consistency), 3-cyclic consistency in VCSPs will no doubt almost always be applied in conjunction with full directional arc consistency.

5. In-scope order-3 irreducibility

We will now demonstrate the importance of 3-cyclic consistency by showing that, in the absence of $\top$-valuations, 3-cyclic consistency implies a local form of optimality.

Definition 5.1. A VCSP(sm) is *finitely-bounded* if $\forall C(P) \in \mathcal{C}\ \forall x \in A_i\ \phi_P(x) < \top$.

All instances of MAX-CSP are finitely-bounded, since all penalties are either 0 or 1, and hence never infinite. In a finitely-bounded VCSP(sm), no tuple is completely inconsistent.

Lemma 5.2. *Any equivalence-preserving transformation from a binary finitely-bounded* VCSP(sm) V *to another binary* VCSP(sm) V' *is equivalent to a set of arc consistency operations* (*i.e., shifting of weights between unary and binary constraints*).

Proof. See Appendix A. □

The importance of Lemma 5.2 will become apparent when we restrict our attention to equivalence-preserving transformations on 3-variable subproblems. Indeed, any set of arc consistency operations on 3 variables is a CCO, since any 3 variables form a simple cycle.

Definition 5.3. Let $k \geqslant 2$. A VCSP is *in-scope* (k, f_{MIN})*-irreducible* if $\forall J \subseteq N$ such that $|J| = k$, for all VCSPs V' derived from V by an equivalence-preserving transformation on J and such that V' has the same set of constraint scopes as V, $f_{\mathrm{MIN}}(V) \geqslant f_{\mathrm{MIN}}(V')$.

It is known that a directional arc consistent VCSP(sm) is in-scope $(2, f_{\mathrm{MIN}})$-irreducible [9]. The following theorem characterises in-scope $(3, f_{\mathrm{MIN}})$-irreducibility in the special case of a finitely-bounded VCSP(sm).

Theorem 5.4. *Let* $V = \langle N, D, \mathcal{C}, S\rangle$ *be a finitely-bounded binary* VCSP(sm). V *is in-scope* $(3, f_{\mathrm{MIN}})$*-irreducible iff* V *is* 3*-cyclic consistent.*

Proof. $(\Rightarrow)$ Trivial.

$(\Leftarrow)$ Let V be a finitely-bounded binary VCSP(sm), and suppose that V is not in-scope $(3, f_{\mathrm{MIN}})$-irreducible. Then there is an equivalence-preserving transformation on some $\{i, j, k\}$, where $i < j < k$, which transforms V into some V', such that $f_{\mathrm{MIN}}(V') > f_{\mathrm{MIN}}(V)$. The minimum valuation in ϕ_i may decrease when V is transformed into V', provided that this decrease is compensated for by a greater aggregate increase in the minimum valuations in the other constraint functions $\phi_j, \phi_k, \phi_{ij}, \phi_{jk}, \phi_{ki}$. However, establishing full directional arc consistency on $V'(\{i, j, k\})$ necessarily produces an equivalent VCSP V'' in which the minimum valuations in the constraint functions $\phi_j, \phi_k, \phi_{ij}, \phi_{jk}, \phi_{ki}$ are all $\bot$ and in which the minimum valuation in ϕ_i is $f_{\mathrm{MIN}}(V'') \geqslant f_{\mathrm{MIN}}(V') > f_{\mathrm{MIN}}(V)$. To prove Theorem 5.4, it is sufficient to express the transformation from V to V'' as an i-increasing CCO on $\{i, j, k\}$ which creates no new constraint. Since we have just shown that it is i-increasing, and by virtue of Theorem 4.4, it only remains to show that the transformation from V to V'' can always be expressed as a CCO. But, by

Lemma 5.2, any equivalence-preserving transformation on three variables is a CCO since it is the result of a set of arc consistency operations on the cycle of variables $\{i, j, k\}$. □

Unfortunately, Theorem 5.4 does not generalise to order $k > 3$. From Lemma 5.2, we know that an equivalence-preserving transformation on $\{i_1, \dots, i_k\}$ is a set of arc consistency operations on the complete graph K_k with nodes $i_1, \dots, i_k$. A CCO is a set of arc consistency operations on a simple cycle of variables. But K_k is only a simple cycle when $k = 3$, and hence, for $k > 3$, there are some equivalence-preserving transformations which are not equivalent to CCOs.

6. Checking cyclic consistency

The results in this section show that cyclic consistency can be checked by solving a certain number of instances of HORNSAT. One consequence of this result is that 3-cyclic consistency can be checked in polynomial time.

Theorem 6.1. *Let $V = \langle N, D, \mathcal{C}, \langle E, \oplus, \geqslant\rangle\rangle$ be a binary* VCSP(sm) *with $I = \{i_1, i_2, \dots, i_r\} \subseteq N$, $i_1 < i_2 < \cdots < i_r$ and $h \in I$. If an (h, x_0)-increasing cyclic consistency operation (CCO) exists on I in V, then an (h, x_0)-increasing CCO exists on I in which all the valuations $d_i(x), u_i(x)$ $(i \in I, x \in A_i)$ lie in $\{\bot, \Delta\}$ for some $\Delta > \bot$.*

Notation. For $\Delta \in E'$ such that $\Delta > \bot$, define the function $f_\Delta : E' \to E'$ by

$$f_\Delta(\alpha) = \Delta \quad \text{if } \alpha \geqslant \Delta;$$
$$f_\Delta(\alpha) = \bot \quad \text{if } \alpha < \Delta.$$

Lemma 6.2. *Let $a, b, c, \Delta \in E'$, $d \in E$ be such that $\Delta > \bot$ and $a < b \Rightarrow b - a \geqslant \Delta$. Then* (a) *$d + a \geqslant b$ implies that $d + f_\Delta(a - c) \geqslant f_\Delta(b - c)$, and* (b) *$d > \bot$ and $d + a > b$ implies that $d + f_\Delta(a - c) > f_\Delta(b - c)$.*

Proof. (a) If $f_\Delta(a - c) \geqslant f_\Delta(b - c)$ then $d + f_\Delta(a - c) \geqslant f_\Delta(b - c)$ since $d \geqslant \bot$. If $f_\Delta(a - c) < f_\Delta(b - c)$ then $a - c < b - c$. Therefore, $a < b$, and hence $b - a \geqslant \Delta$. Thus, by hypothesis, $d \geqslant b - a \geqslant \Delta$. But $f_\Delta(a - c) < f_\Delta(b - c)$ means $(f_\Delta(a - c), f_\Delta(b - c)) = (\bot, \Delta)$, so again $d + f_\Delta(a - c) \geqslant f_\Delta(b - c)$.

(b) If $f_\Delta(a - c) \geqslant f_\Delta(b - c)$ then $d + f_\Delta(a - c) > f_\Delta(b - c)$, since $d > \bot$. If $f_\Delta(a - c) < f_\Delta(b - c)$ then $b - a \geqslant \Delta$, as in the proof of (a), and, by hypothesis, $d > b - a \geqslant \Delta$. Hence, again we have $d + f_\Delta(a - c) > f_\Delta(b - c)$. □

Proof of Theorem 6.1. Let $\mu_h = M(\phi_h)$ and let $\{d_i, u_i\}$ be an (h, x_0)-increasing CCO. Then $\phi_h(x_0) = \mu_h \wedge d_h(x_0) > u_h(x_0)$. Let $\Delta = \min(S)$, where

$$S = \{d_h(x_0) - u_h(x_0)\} \cup \{u_i(x) - d_i(x)\colon i \in I \wedge x \in A_i \wedge u_i(x) > d_i(x)\}$$
$$\cup \{d_i(x) - u_j(y)\colon (i, j) \in \{(i_1, i_2), (i_2, i_3), \dots, (i_r, i_1)\} \wedge x \in A_i \wedge y \in A_j \wedge d_i(x) > u_j(y)\}.$$

Define, for all $i \in I$, for all $x \in A_i$, $D_i(x) = f_\Delta(d_i(x) - u_h(x_0))$ and $U_i(x) = f_\Delta(u_i(x) - u_h(x_0))$. It remains to prove that the weights $D_i(x), U_i(x)$, for $i \in I$ and $x \in A_i$, represent a legal (h, x_0)-increasing CCO, i.e., that conditions (1), (2), (3) of Definition 4.1 are satisfied.

Condition (1): Let $i \in I$ and $x \in A_i$. Lemma 6.2, with $a = d_i(x)$, $b = u_i(x)$, $c = u_h(x_0)$, $d = \phi_i(x)$ and Δ as defined above, tells us that $\phi_i(x) + D_i(x) \geqslant U_i(x)$, since $\phi_i(x) + d_i(x) \geqslant u_i(x)$.

Condition (2): Let $(i, j) \in \{(i_1, i_2), (i_2, i_3), \ldots, (i_r, i_1)\}$, $x \in A_i$ and $y \in A_j$. Lemma 6.2 with $a = u_j(y)$, $b = d_i(x)$, $c = u_h(x_0)$, $d = \phi_{ij}(x, y)$ tells us that $\phi_{ij}(x, y) + U_j(y) \geqslant D_i(x)$, since $\phi_{ij}(x, y) + u_j(y) \geqslant d_i(x)$.

Condition (3): Let $x \in A_h$. Lemma 6.2 with $a = d_h(x)$, $b = u_h(x)$, $c = u_h(x_0)$, $d = \phi_h(x) - \mu_h$ tells us that $\phi_h(x) + D_h(x) \geqslant \mu_h + U_h(x)$ is a consequence of $\phi_h(x) + d_h(x) \geqslant \mu_h + u_h(x)$ and that $\phi_h(x) + D_h(x) > \mu_h + U_h(x)$ is a consequence of $\phi_h(x) + d_h(x) > \mu_h + u_h(x)$. By definition of Δ, $D_h(x_0) = \Delta$ and clearly $U_h(x_0) = \bot$. Therefore $\phi_h(x_0) + D_h(x_0) = \mu_h + \Delta > \mu_h + U_h(x_0)$. □

Definition 6.3. A valuation $\alpha \in E$ is *divisible* if $\exists \delta \in E'$ such that $\delta > \bot$ and $\delta + \delta \leqslant \alpha$.

For example, in the valuation structure $\langle \mathbb{N} \cup \{\infty\}, +, \geqslant \rangle$, 1 is not divisible.

Theorem 6.4. *Let $V = \langle N, D, \mathcal{C}, \langle E, \oplus, \geqslant \rangle \rangle$ be a binary* VCSP(sm) *and $I = \{i_1, i_2, \ldots, i_r\} \subseteq N$, $i_1 < i_2 < \cdots < i_r$ and $h \in I$. The existence of an (h, x_0)-increasing CCO on I in V can be checked in* $O(a^2 r)$ *time, where* $a = \max\{|A_i|: i \in N\}$.

Proof. By Theorem 6.1, it suffices to check whether an (h, x_0)-increasing CCO exists whose valuations $D_i(x), U_i(x)$ $(i \in I,\ x \in A_i)$ all lie in $\{\bot, \Delta\}$ for some $\Delta > \bot$. We will express this problem as an instance of HORNSAT with the following $2ar$ variables: $e_i(x)$ representing $D_i(x) > \bot$ and $v_i(x)$ representing $U_i(x) > \bot$. Let C be the set of clauses constructed as follows.

For all $i \in I - \{h\}$, $x \in A_i$: if $\phi_i(x) = \bot$, then add to C the clause $e_i(x) \vee \neg v_i(x)$ (to code $D_i(x) \geqslant U_i(x)$).

Similarly, for all $(i, j) \in \{(i_1, i_2), (i_2, i_3), \ldots, (i_r, i_1)\}$, $x \in A_i$ and $y \in A_j$: if $\phi_{ij}(x, y) = \bot$, then add to C the clause $v_j(y) \vee \neg e_i(x)$ (to code $U_j(y) \geqslant D_i(x)$).

For all $x \in A_h - \{x_0\}$, if $\phi_h(x) = \mu_h$ or if $\phi_h(x) - \mu_h > \bot$ is not divisible, then add to C the clause $e_h(x) \vee \neg v_h(x)$ (to code $D_h(x) \geqslant U_h(x)$). For x_0, add to C the clauses $\neg v_h(x_0)$, $e_h(x_0)$ (to code $D_h(x_0) > U_h(x_0)$).

By construction of C, if an (h, x_0)-increasing CCO exists whose valuations $D_i(x)$, $U_i(x)$ $(i \in I, x \in A_i)$ all lie in $\{\bot, \Delta\}$ for some $\Delta > \bot$, then there is a solution to C, defined as follows: $e_i(x) =$ true iff $D_i(x) > \bot$ and $v_i(x) =$ true iff $U_i(x) > \bot$. That this satisfies all the clauses in C follows directly from Definition 4.1 of a CCO, except for the clauses associated with valuations $\phi_h(x) - \mu_h > \bot$ which are not divisible. Suppose that $x \in A_h - \{x_0\}$ is such that $\phi_h(x) - \mu_h > \bot$ is not divisible. Now $\phi_h(x) + D_h(x) > \mu_h + U_h(x)$ means that it is impossible that $D_h(x) < U_h(x)$ since, in this case, $\phi_h(x) - \mu_h > U_h(x) - D_h(x) = \Delta$ and hence $\delta + \delta \leqslant \phi_h(x) - \mu_h$ where $\delta = \min(\Delta, \phi_h(x) - \mu_h - \Delta)$.

Suppose, on the other hand, that C has a solution $e_i(x)$, $v_i(x)$ $(i \in I, x \in A_i)$. Then there is an (h, x_0)-increasing CCO defined as follows. For each $i \in I - \{h\}$ and $x \in A_i$, such that $\neg e_i(x) \wedge v_i(x)$, set $\delta_i(x) = \phi_i(x)$; for each $(i, j) \in \{(i_1, i_2), (i_2, i_3), \ldots, (i_r, i_1)\}$, $x \in A_i$ and $y \in A_j$, such that $\neg v_j(y) \wedge e_i(x)$, set $\delta_{ij}(x, y) = \phi_{ij}(x, y)$; for each $x \in A_h - \{x_0\}$, such that $\neg e_h(x) \wedge v_h(x)$, choose $\delta_h(x) > \perp$ so that $\delta_h(x) + \delta_h(x) \leqslant \phi_h(x) - \mu_i$ (this is possible since $\phi_h(x) - \mu_i$ is divisible). Set all other values of δ_i and δ_{ij} (not covered by these cases) to an arbitrary valuation $\lambda \in E$ satisfying $\lambda > \perp$. Let $\Delta = \min(S)$, where

$$S = \{\delta_i(x)\colon\ i \in I \wedge x \in A_i\} \cup \{\delta_{ij}(x, y)\colon\ (i, j) \in \{(i_1, i_2), (i_2, i_3), \ldots, (i_r, i_1)\} \wedge x \in A_i \wedge y \in A_j\}.$$

For each $i \in I$ and $x \in A_i$: define $D_i(x) = \Delta$ if $e_i(x) = \text{true}$, $D_i(x) = \perp$ if $e_i(x) = \text{false}$; define $U_i(x) = \Delta$ if $v_i(x) = \text{true}$, $U_i(x) = \perp$ if $v_i(x) = \text{false}$. It is easy to verify that this is a valid (h, x_0)-increasing CCO.

All clauses in C are Horn clauses. The instance C of HORNSAT can be solved in $O(a^2 r)$ time, since there are $O(a^2 r)$ clauses [11]. □

The following lemma is essential for us to be able to build h-increasing CCO's "brick by brick" as a sequence of (h, x_0)-increasing CCO's.

Lemma 6.5. *Let $V = \langle N, D, \mathcal{C}, \langle E, \oplus, \geqslant\rangle\rangle$ be a binary* VCSP(sm), *$I \subseteq N$ and $h = \min(I)$. Let V' (with constraint functions ϕ') be the result of applying to V a CCO on I. If $M(\phi'_h) = M(\phi_h)$, then V is cyclic consistent on I iff V' is cyclic consistent on I.*

Proof. Let $\{D_i, U_i\colon\ i \in I\}$ be the CCO on I which transforms V to V'. The result of applying any CCO $\{d_i, u_i\colon\ i \in I\}$ on I in V is obviously equivalent to the result of applying the CCO $\{d'_i, u'_i\colon\ i \in I\}$ on I in V', where

$$\forall i \in I\ \forall x \in A_i \quad d'_i(x) = d_i(x) - D_i(x),$$
$$\forall i \in I\ \forall x \in A_i \quad u'_i(x) = u_i(x) - U_i(x).$$

Furthermore, $\{d_i, u_i\colon\ i \in I\}$ is an h-increasing CCO on I in V iff $\{d'_i, u'_i\colon\ i \in I\}$ is an h-increasing CCO on I in V'. The result follows immediately from Definition 4.3 of cyclic consistency. □

Theorem 6.6. *Let $V = \langle N, D, \mathcal{C}, \langle E, \oplus, \geqslant\rangle\rangle$ be a binary* VCSP(sm) *and $I \subseteq N$. Cyclic consistency on I in V can be checked in* $O(a^3 r)$ *time, where $a = \max\{|A_i|\colon\ i \in N\}$ and $r = |I|$.*

Proof. The function **CC**, below, searches for and applies (h, x_0)-increasing CCO's for each $x_0 \in A_h$ such that $\phi_h(x_0) = \mu_h$, until either $M(\phi_h)$ increases (i.e., an h-increasing CCO has been found) or no (h, x_0)-increasing CCO exists. In the latter case, clearly no h-increasing CCO exists, and hence Lemma 6.5 tells us that the original VCSP V was cyclic consistent on I.

In **CC**, $V(I)$ represents the subproblem of V on I, i.e., the constraint functions ϕ_i $(i \in I)$ and ϕ_{ij} $(i, j \in I)$. The instruction *old* $V(I) := V(I)$ makes a copy of the original constraint functions in *old* $V(I)$.

CC(I):
  old $V(I) := V(I)$; $h := \min(I)$; $\mu :=$ MIN$\{\phi_h(x)\colon x \in A_h\}$;
  for each $x_0 \in A_h$ do
  begin if $\phi_h(x_0) = \mu$
    then if an (h, x_0)- increasing CCO exists on I
      then apply this CCO to V;
      else begin $V(I) :=$ *old* $V(I)$;
          {undo updates since *old* V was already cyclic consistent on I}
        return true;
      end;
  end;
  return false;

The time complexity of **CC**(I) is clearly $O(a^3r)$, since, from Theorem 6.4, we know that the search for an (h, x_0)-increasing CCO is $O(a^2r)$ for each $x_0 \in A_h$. □

The following result is an immediate corollary of Theorem 6.6.

Corollary 6.7. *In a binary* VCSP(sm), 3-*cyclic consistency can be checked in* $O(n^3a^3)$ *time.*

7. Establishing in-scope 3-irreducibility in MAX-CSP

We now show how to establish 3-cyclic consistency, and hence in-scope $(3, f_{\mathrm{MIN}})$-irreducibility (by Theorem 5.4), in a finitely-bounded VCSP, such as MAX-CSP.

Definition 7.1. A VCSP $V = \langle N, D, \mathcal{C}, S\rangle$ is *normalised* if

$$\forall C(P) \in \mathcal{C}, \quad P \neq \emptyset \Rightarrow M(\phi_P) = \bot.$$

The call **Normalise**(N) of the following subroutine transforms a fair VCSP V (see Definition 2.8) into an equivalent normalised VCSP. **Normalise** makes use of a constraint ϕ whose scope is the empty set of variables. The valuation assigned to ϕ is a lower bound on the valuations of all solutions to V. After execution of **Normalise**(N), ϕ is equal to $f_{\mathrm{MIN}}(V)$.

Normalise(I):
  for each $C(P) \in \mathcal{C}$ such that $P \subseteq I \wedge P \neq \emptyset$ do
  begin $\mu :=$ MIN$\{\phi_P(x)\colon x \in L(P)\}$;
    $\phi := \phi \oplus \mu$;
    for each $x \in L(P)$ do $\phi_P(x) := \phi_P(x) \ominus \mu$;
  end;

The following algorithm **IS3I** (In-Scope 3-Irreducibility) establishes in-scope $(3, f_{\mathrm{MIN}})$-irreducibility using cyclic consistency operations, provided V is a finitely-bounded

VCSP(sm). The algorithm **FDAC2**(I) establishes full directional arc consistency in the subproblem on I in $O(a^2|I|^2)$ time [8].

```
IS3I(V):
{Initialisation:}
  Normalise(N);
  FDAC2(N);
  L := list of all 3-variable subsets {i, j, k} of N;
{Propagation:}
  While L ≠ Ø do
    Extract some {i, j, k} from L;
    cyclic_consistent := CC({i, j, k});
    if not cyclic_consistent
    then {Establish cyclic consistency on {i, j, k} in V: }
    begin Repeat Normalise({i, j, k});
                 FDAC2({i, j, k});
                 cyclic_consistent := CC({i, j, k});
          Until cyclic_consistent = true;
          Add to L all 3-variable subsets {i', j', k'} ≠ {i, j, k}
            such that {i, j, k} ∩ {i', j', k'} ≠ Ø;
    end;
  end_while;
```

Theorem 7.2. *If* V *is a finitely-bounded* VCSP(sm)*, then* **IS3I**(V) *establishes in-scope* $(3, f_{\text{MIN}})$*-irreducibility.*

Proof. It is clear that when **IS3I**(V) halts, V is cyclic consistent on all 3-variable subsets $\{i, j, k\}$ of N. It follows from Theorem 5.4 that **IS3I**(V) establishes in-scope $(3, f_{\text{MIN}})$-irreducibility. □

Note that the calls of **FDAC2** in **IS3I** are inessential for the validity of Theorem 7.2. However, it is clearly interesting to simultaneously establish both full directional arc consistency and 3-cyclic consistency.

Proposition 7.3. IS3I *establishes full directional arc consistency.*

Proof. Full directional arc consistency is established during the initialisation phase. It can be destroyed only by a call of **CC**($\{i, j, k\}$) which returns 'false'. A call of **CC**($\{i, j, k\}$) which returns 'true' does not modify the constraints. However, full directional arc consistency is re-established by a call of **FDAC2**($\{i, j, k\}$) following every call of **CC**($\{i, j, k\}$) which returns 'false'. Thus **IS3I** establishes full directional arc consistency. □

Theorem 7.4. *Let* V *be an instance of MAX-CSP with only unary and binary constraints.* **IS3I** *establishes in-scope* $(3, f_{\text{MIN}})$*-irreducibility in* V *in* $O(a^3n^4)$ *time.*

Proof. Let m be the number of iterations of the while loop and m_{inc} the number of iterations of the while loop of **IS3I**(V) during which the lower bound ϕ increases. It was shown in the proof of Theorem 6.6 that each call of **CC**$(\{i, j, k\})$ which does not return 'true' increases $M(\phi_i)$ (where we assume wlog that $i = \min\{i, j, k\}$). This increase is automatically passed on to ϕ by **Normalise**$(\{i, j, k\})$.

Since there are $n(n-1)(n-2)/6$ additions to L during the initialisation phase and at most $3(n-3)(n-2)/2$ additions to L each time ϕ increases, m is bounded above by $n(n-1)(n-2)/6 + m_{\text{inc}} * 3(n-3)(n-2)/2$. Since V is an instance of MAX-CSP, m_{inc} is bounded above by $n(n+1)/2$, because ϕ cannot exceed the total number of unary and binary constraints. Thus $m = \mathrm{O}(n^4)$. The time complexity of $\mathrm{O}(a^3 n^4)$ follows from the $\mathrm{O}(a^3)$ time complexity of **CC**$(\{i, j, k\})$ (Theorem 6.6) and the $\mathrm{O}(a^2)$ time complexity of both **Normalise**$(\{i, j, k\})$ and **FDAC2**$(\{i, j, k\})$ [8]. □

Definition 7.5. The *constraint graph* of a binary VCSP $V = \langle N, D, \mathcal{C}, S\rangle$ is the graph G whose nodes are the variables N and such that, for all $i, j \in N$, G contains the edge (i, j) iff there is a constraint between variables i and j in $\mathcal{C}$.

Definition 7.6. A graph has *d-bounded degree* if each node is adjacent to at most d other nodes.

The constraint graph of certain classic constraint satisfaction problems, such as the line drawing labelling problem [7], have d-bounded degree for some small constant d. For example, we can model the labelling of imperfect line drawings of objects with trihedral vertices as a binary VCSP in which the variables are the junctions in the drawing. In this case, the constraint graph has 3-bounded degree.

Theorem 7.7. *Let V be an instance of MAX-CSP with only unary and binary constraints. If the constraint graph of V has d-bounded degree, then in-scope $(3, f_{\text{MIN}})$-irreducibility can be established in $\mathrm{O}(a^3 d^3 n)$ time.*

Proof. Theorem 4.4 tells us that we only need to consider 3-variable subsets $\{i, j, k\}$ which form connected subgraphs in the constraint graph. Let n_3 be the number of such subsets. Then $n_3 \leqslant d^2 n$, since each variable i is connected to at most d variables j which is, in turn, connected to at most d variables. The number c of constraints is bounded above by $n + (nd/2)$ (n unary constraints and $nd/2$ binary constraints). Furthermore, for a given subset of variables $\{i, j, k\}$, the number n_{int} of subsets $\{i', j', k'\}$ of connected variables which intersect $\{i, j, k\}$ is no more than $6d^2$, since, for example, variable i is connected to at most d variables h and $\{i, h\}$ is connected to at most $2d$ other variables. Following the same argument as in the proof of Theorem 7.4, the time complexity of **IS3I** is, in this case, $\mathrm{O}(a^3(n_3 + c.n_{\text{int}})) = \mathrm{O}(a^3(d^2 n + (n + nd/2).6d^2)) = \mathrm{O}(a^3 d^3 n)$. □

Thus, if the constraint graph of an instance of MAX-CSP has d-bounded degree, for some constant d, then 3-cyclic consistency (and hence in-scope $(3, f_{\text{MIN}})$-irreducibility) can be established in time and space which is linear in n, the number of variables.

8. Discussion

Unfortunately, cyclic consistency operations are not sufficient to establish in-scope $(3, f_{\mathrm{MIN}})$-irreducibility when $\top$-valuations are present in the VCSP. Fig. 6 shows an example. The VCSP V in Fig. 6(a) is 3-cyclic consistent. However, V is equivalent to the VCSP V' in Fig. 6(b) and $f_{\mathrm{MIN}}(V') = 1 > 0 = f_{\mathrm{MIN}}(V)$.

Nonetheless, cyclic consistency operations may still be usefully applied to VCSPs containing $\top$-valuations. Furthermore, in the presence of $\top$-valuations, arc and path consistency operations can be applied to the underlying CSP (see Definition 3.1).

In MAX-CSP, the fact that 3-cyclic consistency is equivalent to in-scope $(3, f_{\mathrm{MIN}})$-irreducibility (Theorem 5.4) would seem to indicate that no stronger form of in-scope 3-consistency exists for MAX-CSP. However, this is not true. Value-level 3-cyclic consistency, defined below, is a stronger form of in-scope 3-consistency which also implies $(3, f_{\mathrm{MIN}})$-irreducibility.

Definition 8.1. A binary VCSP(sm) V is *value-level* 3-*cyclic consistent*, if for all $i, j, k \in N$ such that $i < j < k$, for all $x_0 \in A_i$, no (i, x_0)-increasing CCO exists on $\{i, j, k\}$.

To establish value-level 3-cyclic consistency, we need to apply all (i, x_0)-increasing CCOs, whereas to establish 3-cyclic consistency we only need to apply CCOs which actually increase f_{MIN}.

Theorem 8.2. *If V is a binary* VCSP(sm), *then the value-level* 3-*cyclic consistency of V can be checked in* $O(a^3 n^3)$ *time.*

Proof. This follows immediately from Theorem 6.4. □

It is an open question whether value-level 3-cyclic consistency can be established in polynomial time in MAX-CSP.

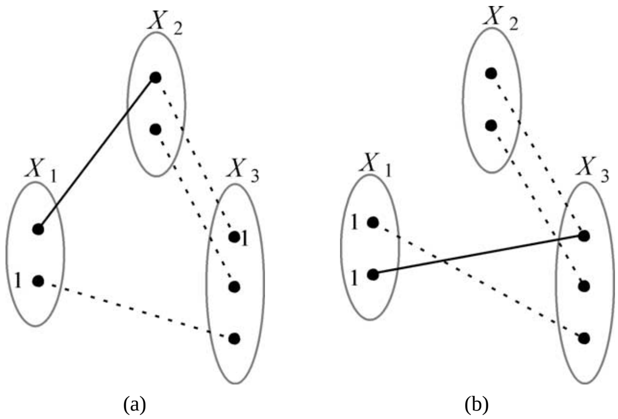

Fig. 6. Two equivalent VCSPs over the valuation structure $\langle \mathbb{N} \cup \{\infty\}, +, \geqslant \rangle$. Full lines have weights of 1 and dotted lines have weights of ∞.

9. Conclusion

3-cyclic consistency is a reduction operation on VCSPs which retains the essential properties of path consistency in crisp constraint satisfaction problems: it performs equivalence-preserving transformations on size-3 subproblems and can be checked in polynomial time. For certain VCSPs, such as MAX-CSP, it can be established in polynomial time and even guarantees a local form of optimality: any 3-cyclic consistency closure attains a local maximum of the natural lower bound f_{MIN} on valuations of solutions.

Several questions remain open. Can we profitably apply cyclic consistency operations dynamically within an intelligent exhaustive search, so that the time spent on cyclic consistency checks is more than compensated by the resulting pruning of the search tree? Are there any tractable classes of valued constraints which can be solved by 3-cyclic consistency?

We have given a polynomial-time algorithm to establish in-scope $(3, f_{\mathrm{MIN}})$-irreducibility in the absence of totally inconsistent valuations. There are two obvious avenues of future research concerning stronger in-scope reduction operations: the search for a polynomial-time algorithm to establish in-scope $(3, f_{\mathrm{MIN}})$-irreducibility in the presence of totally inconsistent valuations and the generalisation of this work to in-scope (k, f_{MIN})-irreducibility, for arbitrary k. It is an open question whether either of these stronger versions of irreducibility can be established in polynomial time.

Acknowledgement

The author would like to thank the anonymous referees whose comments significantly improved the presentation of this paper.

Appendix A

Lemma 5.2. *Any equivalence-preserving transformation from a binary finitely-bounded* VCSP(sm) V *to another binary* VCSP(sm) V' *is equivalent to a set of arc consistency operations* (*i.e., shifting of weights between unary and binary constraints*).

Proof. Let ϕ_i, ϕ_{ij} be the constraint functions of V and ϕ'_i, ϕ'_{ij} the constraint functions of V'. We assume without loss of generality that a constraint exists between each pair of variables $i < j$. For all $(x, y) \in A_i \times A_j$, define

$$\delta_{ij}(u, v) = \phi'_{ij}(u, v) - \phi_{ij}(u, v) \quad \text{if } 1 < i < j;$$

$$\delta_{1j}(u, v) = \phi'_{1j}(u, v) - \phi_{1j}(u, v) + \phi'_j(v) - \phi_j(v) \quad \text{if } 2 < j;$$

$$\delta_{12}(u, v) = \phi'_{12}(u, v) - \phi_{12}(u, v) + \phi'_2(v) - \phi_2(v) + \phi'_1(u) - \phi_1(u).$$

Let $W(W')$ be a version of the VCSP $V(V')$ in which the unary penalties $\phi_j(\phi'_j)$ have been shifted up to the binary constraint $\phi_{1j}(\phi'_{1j})$ for $j > 1$ and to the binary constraint $\phi_{12}(\phi'_{12})$ for $j = 1$. We can consider δ_{ij} as the increase in the constraint function on i, j

caused by the transformation from W to W'. The essential property of the functions δ_{ij} is that $\forall u = (u_1, \ldots, u_n) \in A_1 \times \cdots \times A_n$,

$$\sum_i \sum_{j>i} \delta_{ij}(u_i, u_j) = \mathrm{Val}_{V'}(u) - \mathrm{Val}_V(u) = \perp \tag{A.1}$$

by Definition 2.7 of an equivalence-preserving transformation.

Let $i < j, a, c \in A_i$ and $b, d \in A_j$. Setting $(u_i, u_j) = (a, b), (a, d), (c, b), (c, d)$ in turn in Eq. (A.1), while keeping u_k fixed for all $k \notin \{i, j\}$, allows us to deduce that $\forall a, c \in A_i\ \forall b, d \in A_j$

$$\delta_{ij}(a, b) + \delta_{ij}(c, d) = \delta_{ij}(a, d) + \delta_{ij}(c, b). \tag{A.2}$$

Now, for each $u \in A_i$, let $\alpha_{ij}(u) = \mathrm{MIN}\{\delta_{ij}(u, y)\colon\ y \in A_j\}$ and for each $v \in A_j$, let $\beta_{ij}(v) = \mathrm{MIN}\{\delta_{ij}(x, v) - \alpha_{ij}(x)\colon\ x \in A_i\}$. Consider any $(u, v) \in A_i \times A_j$. Suppose that $y \in A_j$ is such that $\delta_{ij}(u, y) = \alpha_{ij}(u)$ and that $x \in A_i$ is such that $\delta_{ij}(x, v) - \alpha_{ij}(x) = \beta_{ij}(v)$. Then

$$\begin{aligned}
\delta_{ij}(u, v) &= \delta_{ij}(u, y) + \delta_{ij}(x, v) - \delta_{ij}(x, y) \quad \text{by (A.2)} \\
&= \alpha_{ij}(u) + \beta_{ij}(v) + \alpha_{ij}(x) - \delta_{ij}(x, y) \\
&\leqslant \alpha_{ij}(u) + \beta_{ij}(v) \quad \text{by definition of } \alpha_{ij}(x).
\end{aligned}$$

But, by definition, $\beta_{ij}(v) \leqslant \delta_{ij}(u, v) - \alpha_{ij}(u)$ and hence $\delta_{ij}(u, v) \geqslant \alpha_{ij}(u) + \beta_{ij}(v)$. Thus, for $i < j$, $\forall (u, v) \in A_i \times A_j$

$$\delta_{ij}(u, v) = \alpha_{ij}(u) + \beta_{ij}(v). \tag{A.3}$$

Let $i \in N$ and $a, b \in A_i$. Setting $u_i = a, b$ in turn in Eq. (A.1), while keeping u_k fixed for all $k \neq i$, allows us to deduce that $\forall a, b \in A_i$

$$\sum_{j<i} \delta_{ji}(u_j, a) + \sum_{j>i} \delta_{ij}(a, u_j) = \sum_{j<i} \delta_{ji}(u_j, b) + \sum_{j>i} \delta_{ij}(b, u_j)$$

(where i is fixed and each sum is over j). Substituting the values of δ_{ji}, δ_{ij} given by Eq. (A.3) and cancelling,

$$\sum_{j<i} \beta_{ji}(a) + \sum_{j>i} \alpha_{ij}(a) = \sum_{j<i} \beta_{ji}(b) + \sum_{j>i} \alpha_{ij}(b). \tag{A.4}$$

For all $i \in N$, define

$$\sigma_i = \sum_{j<i} \beta_{ji}(a) + \sum_{j>i} \alpha_{ij}(a).$$

By Eq. (A.4), σ_i is independent of the choice of $a \in A_i$. Finally, define, for $i \in N$ and $u \in A_i$, $\delta_i(u) = \phi'_i(u) - \phi_i(u)$.

Consider the following set of arc consistency operations:

(1) for i, j s.t. $i < j$, for $u \in A_i$, shift $\alpha_{ij}(u)$ from $\phi_i(u)$ to $\phi_{ij}(u, v)$ $(v \in A_j)$;
(2) for i, j s.t. $i < j$, for $v \in A_j$, shift $\beta_{ij}(v)$ from $\phi_j(v)$ to $\phi_{ij}(u, v)$ $(u \in A_i)$;
(3) for $j > 1$, for $u \in A_1$, shift σ_j from $\phi_1(u)$ to $\phi_{1j}(u, v)$ $(v \in A_j)$;

(4) for $j > 1$, for $v \in A_j$, shift σ_j from $\phi_{1j}(u, v)$ $(u \in A_1)$ to $\phi_j(v)$;
(5) for $j > 1$, for $v \in A_j$, shift $\delta_j(v)$ from $\phi_{1j}(u, v)$ $(u \in A_1)$ to $\phi_j(v)$;
(6) for $u \in A_1$, shift $\delta_1(u)$ from $\phi_{12}(u, v)$ $(v \in A_2)$ to $\phi_1(u)$.

If W, W' correspond to the versions of V, V' with all weights shifted away from unary constraints, as described above, then operations (1)–(4) correspond to the transformation from W to W', whereas operations (5), (6) correspond to the sum of the transformations $V \to W$ and $W' \to V'$.

Let ψ_i, ψ_{ij} be the constraint functions after applying the set of arc consistency operations (1)–(6) to V. For $1 < i < j$, $\forall (u, v) \in A_i \times A_j$,

$$\psi_{ij}(u, v) = \phi_{ij}(u, v) + \alpha_{ij}(u) + \beta_{ij}(v) = \phi_{ij}(u, v) + \delta_{ij}(u, v) = \phi'_{ij}(u, v).$$

For $j > 2$, $\forall (u, v) \in A_1 \times A_j$,

$$\begin{aligned}\psi_{1j}(u, v) &= \phi_{1j}(u, v) + \alpha_{1j}(u) + \beta_{1j}(v) + \sigma_j - \sigma_j - \delta_j(v)\\ &= \phi_{1j}(u, v) + \delta_{1j}(u, v) - \phi'_j(v) + \phi_j(v)\\ &= \phi'_{1j}(u, v)\end{aligned}$$

and $\forall (u, v) \in A_1 \times A_2$,

$$\begin{aligned}\psi_{12}(u, v) &= \phi_{12}(u, v) + \alpha_{12}(u) + \beta_{12}(v) + \sigma_2 - \sigma_2 - \delta_2(v) - \delta_1(u)\\ &= \phi_{12}(u, v) + \delta_{12}(u, v) - \phi'_2(v) + \phi_2(v) - \phi'_1(u) + \phi_1(u)\\ &= \phi'_{12}(u, v).\end{aligned}$$

For $i > 1$, $\forall u \in A_i$,

$$\begin{aligned}\psi_i(u) &= \phi_i(u) - \sum_{j<i} \beta_{ji}(u) - \sum_{j>i} \alpha_{ij}(u) + \sigma_i + \delta_i(u)\\ &= \phi_i(u) - \sigma_i + \sigma_i + \phi'_i(u) - \phi_i(u)\\ &= \phi'_i(u)\end{aligned}$$

and $\forall u \in A_1$,

$$\begin{aligned}\psi_1(u) &= \phi_1(u) - \sum_{j>1} \alpha_{1j}(u) - \sum_{i>1} \sigma_i + \delta_1(u)\\ &= \phi_1(u) - \sum_i \sigma_i + \phi'_1(u) - \phi_1(u)\\ &= \phi'_1(u) - \sum_i \sum_{j>i} \delta_{ij}(a_i, a_j) \quad \text{for any } a = (a_1, \dots, a_n) \in A_1 \times \cdots \times A_n\\ &= \phi'_1(u) \quad \text{by (A.1)}.\end{aligned}$$

Thus, the transformation from V to V' is equivalent to the set of arc consistency operations (1)–(6). □

References

[1] M.S. Affane, H. Bennaceur, A weighted arc consistency technique for MAX-CSP, in: Proc. of the 13th ECAI, Brighton, UK, 1998, pp. 209–213.
[2] S. Bistarelli, U. Montanari, F. Rossi, Semiring based constraint solving and optimization, J. ACM 44 (2) (1997) 201–236.
[3] S. Bistarelli, H. Fargier, U. Montanari, F. Rossi, T. Schiex, G. Verfaillie, Semiring-based CSPs and valued CSPs: Frameworks, properties and comparison, Constraints 4 (1999) 199–240.
[4] C. Bliek, D. Sam-Haroud, Path consistency on triangulated constraint graphs, in: Proc. IJCAI-99, Stockholm, Sweden, 1999, pp. 456–461.
[5] A. Chmeiss, P. Jégou, Path consistency: when space misses time, in: Proc AAAI-96, Portland, OR, 1996, pp. 196–201.
[6] M.C. Cooper, An optimal k-consistency algorithm, Artificial Intelligence 41 (1989) 89–95.
[7] M.C. Cooper, Linear-time algorithms for testing the realisability of line drawings of curved objects, Artificial Intelligence 108 (1999) 31–67.
[8] M.C. Cooper, Reduction operations in fuzzy and valued constraint satisfaction, Fuzzy Sets and Systems 134 (3) (2003).
[9] M.C. Cooper, T. Schiex, Arc consistency for soft constraints, Artificial Intelligence, in press; doi:10.1016/j.artint.2003.09.002.
[10] R. Debruyne, C. Bessière, From restricted path consistency to max-restricted path consistency, in: G. Smolka (Ed.), Principles and Practice of Constraint Programming—CP'97, in: Lecture Notes in Comput. Sci., vol. 1330, Springer, Berlin, 1997, pp. 312–326.
[11] W.F. Dowling, J.H. Gallier, Linear-time algorithms for testing the satisfiability of propositional Horn formulae, J. Logic Programming 3 (1984) 267–284.
[12] E. Freuder, R. Wallace, Partial constraint satisfaction, Artificial Intelligence 58 (1992) 21–70.
[13] C. Han, C. Lee, Comments on Mohr and Henderson's path consistency algorithm, Artificial Intelligence 36 (1988) 125–130.
[14] J. Larrosa, P. Meseguer, T. Schiex, Maintaining reversible DAC for Max-CSP, Artificial Intelligence 107 (1) (1999) 149–163.
[15] R. Mohr, T.C. Henderson, Arc and path consistency revisited, Artificial Intelligence 28 (1986) 225–233.
[16] T. Schiex, Arc consistency for soft constraints, in: Principles and Practice of Constraint Programming—CP 2000, in: Lecture Notes in Comput. Sci., vol. 1894, Springer, Berlin, 2000, pp. 411–424.
[17] T. Schiex, H. Fargier, G. Verfaillie, Valued constraint satisfaction problems: hard and easy problems, in: Proc. IJCAI-95, Montreal, Quebec, 1995, pp. 631–637.
[18] T. Schiex, J.-C. Régin, C. Gaspin, G. Verfaillie, Lazy arc consistency, in: Proc AAAI-96, Portland, OR, 1996, pp. 216–221.