Soft Constraints Processing over Divisible Residuated Lattices

Simone Bova

Department of Computer Science, University of Milan
Via Comelico 39, 20135 Milan (Italy)
bova@dico.unimi.it

Abstract. We claim that divisible residuated lattices (DRLs) can act as a unifying evaluation framework for soft constraint satisfaction problems (soft CSPs). DRLs form the algebraic semantics of a large family of substructural and fuzzy logics [13, 15], and are therefore natural candidates for this role. As a preliminary evidence in support to our claim, along the lines of Cooper et al. and Larrosa et al. [11, 18], we describe a polynomial-time algorithm that enforces k-hyperarc consistency on soft CSPs evaluated over DRLs. Observed that, in general, DRLs are neither idempotent nor totally ordered, this algorithm accounts as a generalization of available enforcing algorithms over commutative idempotent semirings and fair valuation structures [4, 11].

1 Introduction

A constraint satisfaction problem (CSP) is the problem of deciding, given a collection of constraints on variables, whether or not there is an assignment to the variables satisfying all the constraints. In the *crisp* setting [19], any assignment satisfying all the constraints provides a solution, and any solution is as good as any other. In the *soft* setting [5], more generally, each constraint maps the assignments to a *valuation structure*, which is a bounded poset equipped with a suitable *combination* operator; the task is to find an assignment such that the combination of its images under all the constraints is maximal in the order of the valuation structure. Formal definitions are given in Section 2.

In its general formulation, the soft CSP is NP-complete, so that research effort is currently aimed to characterize tractable cases [7, 6], and investigating constraints processing heuristics; amongst the latter, *enforcing* algorithms are of the foremost importance. [1] A typical enforcing algorithm takes in input a soft CSP and enforces a *local consistency* property over it, producing two possible outcomes: either the input problem is found locally inconsistent, implying its global inconsistency; or else, the input problem is transformed into an *equivalent* problem (called *closure*), possibly inconsistent but *easier*, that is, with a smaller solution space. Despite their incompleteness as inconsistency tests, enforcing algorithms are useful as subprocedures in the exhaustive search for an optimal

[1] For further background on constraint processing, we refer the reader to [12].

solution, for instance in *branch and bound* search. The generalization of local consistency notions and techniques from the crisp to the soft setting plays a central role in the algorithmic investigation of the soft CSP: for this reason, any class of structures that allows for an easy migration of local consistency techniques in the soft setting deserves consideration [4, 18, 11].

Not surprisingly, the *weaker* the properties of the valuation structure are, the harder it is to migrate a local consistency technique from the crisp to the soft setting. Indeed, a crisp CSP is equivalent to a soft CSP over a valuation structure with very *strong* properties: the algebra $(\{0,1\}, \leq, \odot, \bot, \top)$, where $\bot = 0 \leq 1 = \top$ and $x \odot y = 1$ if and only if $x = y = 1$. At the other extreme, the *weakest* possible valuation structure has to be a bounded poset, with top element $\top$ and bottom element $\bot$, equipped with a commutative, associative operation $x \odot y$ which is monotone over the order ($x \leq y$ implies $x \odot z \leq y \odot z$), has $\top$ as identity ($x \odot \top = x$) and $\bot$ as annihilator ($x \odot \bot = \bot$). Intuitively, an assignment mapped to $\top$ by a constraint is entirely satisfactory, and an assignment mapped to $\bot$ is entirely unsatisfactory; if two assignments are mapped to x and y, in case $x \leq y$, the latter is preferred to the former, and in case $x \parallel y$, none is preferred over the other; the operator $\odot$ combines constraints in such a way that adding constraints shrinks the solution space (as boundary cases, $\top$ does not shrink the solution space, and $\bot$ empties the solution space). In this setting, two options arise: whether or not to allow incomparability (formally, whether or not to admit *non-totally* ordered valuation structures); and, whether or not to keep into account repetitions (formally, whether or not to allow for valuation structures with *nonidempotent* combination operators). [2] The aforementioned algebra $(\{0,1\}, \leq, \odot, \bot, \top)$ is strong in the sense that it is totally ordered and idempotent.

In this paper, we propose *(commutative bounded) divisible residuated lattices* (in short, *DRLs*) as a unifying evaluation framework for soft constraints. Despite DRLs form an intensively studied algebraic variety since [22], they have never been proposed as an evaluation framework for soft constraints. However, there are robust motivations for considering DRLs in the soft CSP setting, coming from logic and algebra. As already mentioned, the soft CSP is a generalization of the crisp CSP. Conversely, the crisp CSP can be seen as a particular soft CSP, evaluated over the algebra $(\{0,1\}, \leq, \odot, \bot, \top)$, that is, a reduct of the familiar Boolean algebra **2** (taking $\odot$ as $\wedge$). Since **2** and the meet operation in **2** form the algebraic counterparts of Boolean logic and Boolean conjunction respectively, it is natural to intend the combination operator $\odot$ in a valuation structure as a generalization of the meet operation in **2**, and to investigate the algebraic counterparts of logics that generalize Boolean conjunction as candidate valuation structures for soft CSPs. Intriguingly, a central approach in the area of *mathematical fuzzy logic*, popularized by Hájek [15], relies on the idea of generalizing Boolean logic starting from a generalization of Boolean conjunction by means of a class of functions called *(continuous) triangular norms* [17]. The idea is the following. A triangular norm $*$ is an associative, commutative, continuous binary function

[2] In the idempotent case $x \odot x = x$, so that repetitions do not matter.

over the real interval $[0,1]$; moreover, $*$ is monotone over the (total, dense and complete) order of reals in $[0,1]$, has 1 as identity and 0 as annihilator. Given a (continuous) triangular norm $*$, there exists a unique binary function $\rightarrow_*$ on $[0,1]$ satisfying the *residuation* equivalence $x*z \leq y$ if and only if $z \leq x \rightarrow_* y$, namely, $x \rightarrow_* y = \max\{z \mid x*z \leq y\}$. This function is called *residuum*, and is a generalization of Boolean implication. Corresponding to any triangular norm $*$, a propositional fuzzy logic $L_* = ([0,1], \wedge, \vee, \odot, \rightarrow, \neg, \bot, \top)$ is obtained by interpreting propositional variables over $[0,1]$, $\bot$ over 0, $\odot$ over $*$, $\rightarrow$ over $\rightarrow_*$, and eventually by defining $\neg x = x \rightarrow \bot$, $\top = \neg\bot = 1$, $x \wedge y = x \odot (x \rightarrow y) = \min(x,y)$, and $x \vee y = ((x \rightarrow y) \rightarrow y) \wedge ((y \rightarrow x) \rightarrow x) = \max(x,y)$. Boolean logic can be readily recovered from L_* by restricting the domain and the connectives to $\{0,1\}$. As much as Boolean algebras form the equivalent algebraic semantics of Boolean logic, the variety of *BL-algebras* (defined in Section 3) forms the algebraic semantics of *Hájek's basic logic*, the logic of all continuous triangular norms and their residua [15, 10].

Therefore, BL-algebras can be regarded as a first candidate evaluation framework for soft CSPs. We shall see that, as far as we are concerned with the implementation of local consistency techniques, for instance the k-hyperarc consistency enforcing algorithm presented in Section 4, *prelinearity* turns out to be redundant. Since prelinearity is exactly the property that specializes BL-algebras inside the class of DRLs [16], we are led to the latter as a defensible level of generality for our unifying evaluation framework. On the logical side, the variety of DRLs forms the algebraic semantics of an intersecting common fragment of basic logic and intuitionistic logic, called *generalized basic logic* [3].

We shall observe that DRLs, in general lattice ordered and nonidempotent, "subsume" preeminent valuation structures where local consistency techniques succeeded, namely, commutative idempotent semirings, lattice ordered and idempotent [4], and fair valuation structures, totally ordered and nonidempotent [11]. Compare Proposition 1 and Proposition 2 in Section 3.

As a preliminary, initial evidence in support of the proposal of DRLs as valuation structures for soft CSPs, we shall prove that DRLs readily host a polynomial-time algorithm that enforces a useful local consistency property, called *k-hyperarc consistency* (compare Definition 5 and Theorem 5 in Section 4). This property guarantees that any *consistent* assignment to a variable i extends to an assignment to any other $\leq k-1$ variables constrained by i, without producing additional costs. We insist that our algorithm works *uniformly* over every DRL, including the aforementioned, previously investigated structures as special cases.

DRLs allow for an extensive, smooth migration of constraint processing techniques from the crisp to the soft setting, far beyond the technical result presented in Section 4, which is intended as a first, concrete example of this new research line. For instance, we reasonably expect that the problem of finding efficiently *optimal* closures (in a suitable sense, required to embed enforcing algorithms

into branch and bound exhaustive search) can be formalized in purely algebraic and logical terms in the setting proposed in this paper. [3]

We remark that analogous local consistency techniques have been investigated by Bistarelli and Gadducci over tropical residuated semirings [2], and we encourage a future comparison of the two settings in terms of unifying potential, structural insight, and computational viability. We also remark that the idea of formalizing soft constraints consistency techniques as many-valued logics refutations appears in the work of Ansótegui et al. [1].

Outline. The paper is organized as follows. In Section 2, we define soft CSPs and valuation structures. In Section 3, we define divisible residuated lattices, and we list a number of properties qualifying DRLs as suitable and natural valuation structures for soft constraints. Then, we describe the relation between evaluation frameworks such as commutative idempotent semirings and fair valuation structures, and DRLs. In Section 4 we present the main technical contribution of this paper, that is a uniform polynomial-time algorithm for k-hyperarc consistency enforcing on soft CSPs evaluated over DRLs. For background on partial orders and universal algebra, we refer the reader to any standard reference.

2 Soft Constraint Satisfaction Problems

In this section, we define formally the notions of soft CSPs, valuation structure, and optimal solution to a soft CSP.

A *(soft) constraint satisfaction problem* (in short, *CSP*) is a tuple $\mathbf{P} = (X, D, P, \mathbf{A})$ specified as follows. $X = \{1, \dots, n\} = [n]$ is a set of *variables*, and $D = \{D_i\}_{i \in [n]}$ is a set of finite *domains* over which variables are assigned, variable i being assigned over domain D_i. Let $Y \subseteq X$. We let

$$l(Y) = \prod_{i \in Y} D_i$$

denote all the assignments of variables in Y onto the corresponding domains (*tuples*). If $Y = \emptyset$, then $l(Y)$ contains only the empty tuple. For any $Z \subseteq Y$, we denote by $t|_Z$ the *projection* of t onto the variables in Z. For every $i \in Y$, $a \in D_i$ and $t \in l(Y \setminus \{i\})$, we let $t \cdot a$ denote the tuple t' in $l(Y)$ such that $t'|_{\{i\}} = a$ and $t'|_{Y \setminus \{i\}} = t$ (if $Y = \{i\}$, then $t \cdot a = a$).

$\mathbf{A}$ is an algebra with domain A and signature including a binary relation $\leq$, a binary operation $\odot$ and constants $\top$, $\bot$, such that the reduct $(A, \leq, \top, \bot)$ is a bounded poset (that is, $\leq$ is a partial order with greatest element $\top$ and least element $\bot$), and the reduct $(A, \odot, \top)$ is a commutative monoid (that is, $\odot$ is commutative and associative and has identity $\top$) where $\odot$ is *monotone* over $\leq$, that is $x \leq y$ implies $x \odot z \leq y \odot z$. $\mathbf{A}$ is called the *valuation structure* of $\mathbf{P}$, and $\odot$ is called the *combination* operator over $\mathbf{A}$.

[3] This key problem has been recently solved over fair valuation structures [8]. Another, weaker consistency property, to be investigated in the DRLs setting, is *virtual* consistency [9].

P is a finite multiset [4] of *constraints*. Each constraint $C_Y \in P$ is defined over a subset $Y \subseteq X$ as a map

$$C_Y : \prod_{i \in Y} D_i \to A.$$

A constraint C_Y has *scope* Y and *arity* $|Y|$.

Let $(C_{Y_1}, \ldots, C_{Y_m})$ be an m-tuple of constraints in P, and let f be an m-ary operation on A. Then, $f(C_{Y_1}, \ldots, C_{Y_m})$ is the constraint with scope $Y_1 \cup \cdots \cup Y_m$ defined by putting, for every $t \in l(Y_1 \cup \cdots \cup Y_m)$:

$$f(C_{Y_1}, \ldots, C_{Y_m})(t) = f(C_{Y_1}(t|_{Y_1}), \ldots, C_{Y_m}(t|_{Y_m})).$$

The set $S(\mathbf{P})$ of *(optimal) solutions* to $\mathbf{P}$ is equal to the set of $t \in l(X)$ such that $\bigodot_{C_Y \in P} C_Y(t|_Y)$ is *maximal* in the poset:

$$\left\{ \bigodot_{C_Y \in P} C_Y(t|_Y) \;\middle|\; t \in l(X) \right\} \subseteq A,$$

where an element x is maximal in a poset if there is no element $y > x$ in the poset (notice that maximal elements in a poset form an antichain). If $S(\mathbf{P}) = \{\bot\}$, we say that $\mathbf{P}$ is *inconsistent*.

Let $\mathbf{P} = (X, D, P, \mathbf{A})$ and $\mathbf{P}' = (X, D, P', \mathbf{A})$ be CSPs. We say that $\mathbf{P}$ and $\mathbf{P}'$ are *equivalent* (in short, $\mathbf{P} \equiv \mathbf{P}'$) if and only if for every $t \in l(X)$,

$$\bigodot_{C_Y \in P} C_Y(t|_Y) = \bigodot_{C_Y \in P'} C_Y(t|_Y).$$

In particular, if $\mathbf{P} \equiv \mathbf{P}'$, then $S(\mathbf{P}) = S(\mathbf{P}')$.

In the sequel we shall assume the following, without loss of generality: P contains at most one constraint with scope $Y \neq \emptyset$ for every $Y \subseteq X$ (otherwise, we replace any pair of constraints C'_Y, C''_Y by the constraint C_Y defined by $C_Y(t|_Y) = C'_Y(t|_Y) \odot C''_Y(t|_Y)$ for every $t \in l(Y)$); P contains all the constraints $C_{\{i\}}$ for $i = 1, \ldots, n$ (otherwise, we add the constraint $C_{\{i\}}$ stipulating that $C_{\{i\}}(a) = \top$ for every $a \in D_i$); $C_{\{i\}}(a) > \bot$ for every $a \in D_i$ (otherwise, we remove a from D_i, declaring the problem inconsistent if D_i becomes empty). Moreover, we shall assume that constraints are implemented as tables, such that entries can be both retrieved and modified, and that algebraic operations over the valuation structure are polynomial-time computable in the size of their inputs.

3 Divisible Residuated Lattices

In this section, we introduce the variety of DRLs and some of its subvarieties, which are interesting with respect to soft CSPs. We give the logical interpretation of each mentioned algebraic variety, and we formalize the relation between DRLs and, commutative idempotent semirings and fair valuation structures.

[4] Multisets are necessary to support nonidempotent combinations of constraints.

Definition 1 (Divisible Residuated Lattice, DRL). *A* divisible residuated lattice [5] *is an algebra* $(A, \vee, \wedge, \odot, \to, \top, \bot)$ *such that:* (i) $(A, \odot, \top)$ *is a commutative monoid;* (ii) $(A, \vee, \wedge, \top, \bot)$ *is a bounded lattice (we write* $x \leq y$ *if and only if* $x \wedge y = x$*);* (iii) residuation *holds, that is,* $x \odot z \leq y$ *if and only if* $z \leq x \to y$*;* (iv) divisibility *holds, that is,* $x \wedge y = x \odot (x \to y)$*. A DRL is called a* DRL-chain *if its lattice reduct is totally ordered.*

We remark that residuation can be readily rephrased in equational terms, so that DRLs form a variety. Notice that divisible residuated chains are not closed under direct products, thus they do not form a variety. As a matter of fact, the lattice reduct of a DRL is *distributive*, that is, $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$.

The monoidal operation of a DRL matches the minimal requirements imposed over the combination operator of a valuation structure in Section 2, as summarized by the following fact [14].

Fact 1 (DRLs Basic Properties) *Let* $\mathbf{A}$ *be a DRL. For every* $x, y, z \in A$*:* (i) $x \odot (y \odot z) = (x \odot y) \odot z$*,* $x \odot y = y \odot x$*,* $x \odot \top = x$*, and* $x \odot \bot = \bot$*;* (ii) $x \leq y$ *implies* $x \odot z \leq y \odot z$ *(in particular,* $x \odot x \leq x$*).*

We exploit the following facts from [14].

Fact 2 (DRLs Extra Properties) *Let* $\mathbf{A}$ *be a DRL. For every* $x, y, z \in A$*:* (i) $x \leq y$ *if and only if* $x \to y = \top$*;* (ii) $y \leq x$ *implies* $x \odot (x \to y) = y$*;* (iii) $y \leq z$ *implies* $(x \odot z) \odot (z \to y) = x \odot y$*;* (iv) $x \odot (y \vee z) = (x \odot y) \vee (x \odot z)$*.*

Fact 3 *[20] Let* $(A, \vee, \wedge, \top, \bot)$ *be a complete bounded lattice, and let* $\odot$ *be a commutative monotone* [6] *operation over* A *such that* $\odot$ *distributes over* $\vee$*. There exists a unique operation* $x \to y$ *satisfying residuation, namely,* $x \to y = \bigvee\{z \mid x \odot z \leq y\}$*.*

In the rest of this section, we discuss the relation between commutative idempotent semirings and fair valuation structures, and DRLs. We first introduce some subvarieties of DRLs.

Definition 2 (DRLs Subvarieties). *A* BL-algebra *is a DRL satisfying* prelinearity*, that is,* $(x \to y) \vee (y \to x) = \top$*. A* Heyting algebra *is a DRL satisfying* idempotency*, that is,* $x \odot x = x$*. A* Gödel algebra *is an idempotent BL-algebra. A Heyting algebra (or a Gödel algebra) is a* Boolean algebra *if it satisfies* involutiveness*, that is,* $\neg\neg x = x$ *where* $\neg x = x \to \bot$*.*

As we mentioned in the introduction, the variety of BL-algebras form the equivalent algebraic semantics of Hájek's basic logic. Analogously, the varieties of Heyting algebras, Gödel algebras, and Boolean algebras respectively, form the equivalent algebraic semantics of *intuitionistic logic*, *Gödel logic*, and *classical logic* [21, 15].

[5] To our aims, we can restrict to commutative and bounded residuated lattices. We refer the reader to [16] for a general definition.

[6] Monotonicity of $\odot$ on both arguments is sufficient.

We consider first commutative idempotent semirings. The restriction to the idempotent case is motivated in this context, since local consistency techniques succeed only over idempotent semirings [5].

Definition 3 (Commutative Idempotent Semiring, CIS). *A* commutative idempotent semiring *is an algebra* $(A, \vee, \odot, \top, \bot)$ *such that:* (i) $\vee$ *is commutative, associative, idempotent,* $x \vee \bot = x$ *and* $x \vee \top = \top$*;* (ii) $\odot$ *is commutative, associative, idempotent,* $x \odot \top = x$ *and* $x \odot \bot = \bot$*;* (iii) $\odot$ *distributes over* $\vee$*, that is* $x \odot (y \vee z) = (x \odot y) \vee (x \odot z)$*.*

Fact 4 *[4, Theorem 2.9, Theorem 2.10] Let* $\mathbf{A} = (A, \vee, \odot, \top, \bot)$ *be a CIS. Then,* $(A, \vee, \wedge, \top, \bot)$*, where* $x \wedge y = x \odot y$*, is a complete bounded distributive lattice.*

Proposition 1. *Let* $\mathbf{A} = (A, \vee, \odot, \top, \bot)$ *be a CIS. Then, the expansion* $\mathbf{A}' = (A, \vee, \wedge, \odot, \rightarrow, \top, \bot)$ *of* $\mathbf{A}$*, where* $x \wedge y = x \odot y$ *and* $x \rightarrow y = \bigvee\{z \mid x \odot z \leq y\}$*, is a Heyting algebra.*

Proof. It is sufficient to prove that $(A, \vee, \wedge, \top, \bot)$ is a bounded distributive lattice, and that $\rightarrow$ is the residuum of $\wedge$. The first part is given by Fact 4. The second part is given by Fact 3: indeed, $(A, \vee, \odot, \top, \bot)$ is complete by Fact 4, $\odot$ is monotone by [4, Theorem 2.4], and $\odot$ distributes over $\vee$ by Definition 3(iii), hence the operation $\rightarrow$ is the uniquely determined residuum of $\wedge$. □

Next we consider fair valuation structures. Accordingly to [11], a fair valuation structure is a structure $(A, \leq, \oplus, \ominus, \top, \bot)$ such that $(A, \leq, \top, \bot)$ is a bounded chain, the combination operator $\oplus$ is commutative, associative, monotone, with identity $\bot$ and annihilator $\top$, and the structure is *fair*, that is, for every $x \leq y \in A$ there exists a maximum $z \in A$, denoted by $y \ominus x$, such that $x \oplus z = y$. The fairness property is crafted ad hoc to preserve the soundness of constraints processing inside the adopted nonidempotent framework [11, Section 4]. Technically, the authors have to guarantee that $z \leq y$ implies $x \oplus y = (x \oplus z) \oplus (y \ominus z)$. We propose here a different, dual definition of a fair valuation structure.

Definition 4 (Dual Fair Valuation Structure, FVS). *A* (dual) fair valuation structure *is an algebra* $\mathbf{A} = (A, \vee, \wedge, \odot, \rightarrow, \top, \bot)$ *such that* $(A, \vee, \wedge, \top, \bot)$ *is a bounded chain,* $(A, \odot, \top)$ *is a commutative monoid, and* $\mathbf{A}$ *satisfies residuation and divisibility.*

Remarkably, the aforementioned technical condition, which in our setting becomes $y \leq z$ implies $x \odot y = (x \odot z) \odot (z \rightarrow y)$, holds by divisibility. The proposed dualization is defensible in logical terms, since the operation of combining soft constraints is intended as a *conjunction*, and the monoidal operation of a DRL is in fact a generalization of Boolean conjunction. In [11], the authors explicitly relate their combination operator with *triangular conorms*. The latter operations, dual to triangular norms, are customarily intended as generalizations of Boolean *disjunction*.

Proposition 2. *A FVS is a DRL-chain.*

We conclude this section mentioning that the soft CSP evaluation framework known as *fuzzy CSP* [4], which has the form $([0,1], \vee, \wedge, \top, \bot)$, can be extended to the Gödel chain $([0,1], \vee, \wedge, \odot, \to, \top, \bot)$ putting $x \odot y = x \wedge y$ and $x \to y$ equal to y if $y > x$ and to $\top$ otherwise. This chain singly generates the variety of Gödel algebras.

4 Enforcing k-Hyperarc Consistency on Soft CSPs over Divisible Residuated Lattices

In this section, we define a property of local consistency, called k-hyperarc consistency, and we describe a polynomial-time algorithm that enforces k-hyperarc consistency on soft CSPs evaluated over DRLs. Syntactically, the pseudocode is almost identical to that presented in [11, 18]; the important and nontrivial point here is to show that it is sound over weaker structures, namely, DRLs (Lemma 2). [7]

Definition 5 (k-Hyperarc Consistency). *Let $\mathbf{P} = (X, D, P, \mathbf{A})$ be a CSP. Let $Y \subseteq X$ such that $2 \leq |Y| \leq k$ and $C_Y \in P$. We say that Y is* k-hyperarc consistent *if for each $i \in Y$ and each $a \in D_i$ such that $C_{\{i\}}(a) > \bot$, there exists $t \in l(Y \setminus \{i\})$ such that,*

$$C_{\{i\}}(a) = C_{\{i\}}(a) \odot C_Y(t \cdot a). \tag{1}$$

We say that $\mathbf{P}$ is k-hyperarc consistent *if every $Y \subseteq X$ such that $2 \leq |Y| \leq k$ and $C_Y \in P$ is k-hyperarc consistent.*

Notice that equation (1) holds if $C_Y(t \cdot a) = \top$. In words, Y is k-hyperarc consistency if each assignment $a \in D_i$ of variable $i \in Y$ such that $C_{\{i\}}(a) > \bot$, extends to an assignment $t \in l(Y \setminus \{i\})$ of variables $Y \setminus \{i\}$ without producing additional costs.

The idea beyond enforcing algorithms is to explicitate implicit constraints induced by the problem over certain subsets of variables, thus possibly discovering a *local* unsatisfiability at the level of these variables. As a specialization of this strategy, our algorithm shifts costs from constraints of arity greater than one to constraints of arity one, thus it possibly reveals the unsatisfiability of the subproblem induced over a single variable (or else, it possibly shrinks the domain of that variable). Such a local unsatisfiability implies the unsatisfiability of the whole problem, as the following proposition shows.

Proposition 3. *Let $\mathbf{P} = (X, D, P, \mathbf{A})$ be a CSP and let $i \in [n]$ be such that $C_{\{i\}} \in P$ and $C_{\{i\}}(a) = \bot$ for every $a \in D_i$. Then, $\mathbf{P}$ is inconsistent.*

[7] A technical advance of the present procedure, compared with the analogous procedure presented by Bistarelli and Gadducci over tropical residuated semirings [2], is termination.

Proof. First recall that for every $x \in A$ it holds that $x \odot \bot = \bot$. But then, $C_{\{i\}}(t|_{\{i\}}) = \bot$ for every $t \in l(X)$, so that $\bigodot_{C_Y \in P} C_Y(t|_Y) = \bot$. Therefore, $S(\mathbf{P}) = \{\bot\}$ and $\mathbf{P}$ is inconsistent. □

Algorithm: k-HyperarcConsistency
Input: A CSP $\mathbf{P} = (X, D, P, \mathbf{A})$.
Output: $\bot$ or a k-hyperarc consistent CSP $\mathbf{P}' = (X, D, P', \mathbf{A})$ equivalent to $\mathbf{P}$.

```
k-HyperarcConsistency((X, D, P, A))
1   Q ← {1, ..., n}
2   while Q ≠ ∅ do
3     i ← Pop(Q)
4     foreach Y ⊆ X such that 2 ≤ |Y| ≤ k, i ∈ Y and C_Y ∈ P do
5       domainShrinks ← Project(Y, i)
6       if C_{i}(a) = ⊥  for each  a ∈ D_i then
7         return ⊥
8       else if domainShrinks then
9         Push(Q, i)
10      endif
11    endforeach
12  endwhile
13  return (X, D, P', A)

Project(Y, i)
14    domainShrinks ← false
15    foreach a ∈ D_i such that C_{i}(a) > ⊥ do
16      x ← a maximal element in {C_Y(t · a) | t ∈ l(Y \ {i})}
17      C_{i}(a) ← C_{i}(a) ⊙ x
18      if C_{i}(a) = ⊥ then
19        domainShrinks ← true
20      endif
21      foreach t ∈ l(Y \ {i}) do
22        C_Y(t · a) ← (x → C_Y(t · a))
23      endforeach
24    endforeach
25    return domainShrinks
```

As already discussed in the introduction, enforcing k-hyperarc consistency over the k-hyperarc inconsistent problem $\mathbf{P}$ may return in output several distinct k-hyperarc consistent problems, depending on the choices made on Lines 1, 3, 4 and 16.

In the rest of this section, we prove that the algorithm runs in polynomial-time (Lemma 1) and is sound (Lemma 2), leading to our main technical result (Theorem 5).

Lemma 1 (Complexity). *Let* $\mathbf{P} = (X, D, P, \mathbf{A})$ *be a CSP, where* $X = [n]$, $d = \max_{i \in [n]} |D_i|$ *and* $e = |P|$. *Then,* k-HyperarcConsistency *terminates in at most* $O(e^2 \cdot d^{k+1})$ *time.*

Proof. The main loop in Lines 2-12 iterates at most $n(d+1) \leq e(d+1)$ times, since $n \leq e$ without loss of generality and each $i \in [n]$ is added to Q once on Line 1 and at most d times on Line 9 (once for each shrink of domain D_i of size $\leq d$). Each iteration of the main loop involves at most e iterations of the loop nested in Lines 4-11, since there are at most e constraints satisfying the condition in Line 4 with respect to any given $i \in [n]$. Each such nested iteration amounts to an invocation of Project and an iteration over domain D_i of size $\leq d$. Any invocation of Project amounts to an iteration over domain D_i of size $\leq d$ on Line 15, and for each such iteration, two iterations over all the $\leq d^{k-1}$ tuples $t \in l(Y \setminus \{i\})$, observing that $1 \leq |Y \setminus \{i\}| \leq k-1$ (Line 16 and Lines 21-23). Summarizing, the algorithm executes at most

$$(e(d+1))e(d+d(2d^{k-1}))$$

many iterations, so that it terminates in at most $O(e^2 \cdot d^{k+1})$ time. □

Lemma 2 (Soundness). *Let* $\mathbf{P} = (X, D, P, \mathbf{A})$ *be a CSP, and consider the output of* k-HyperarcConsistency($\mathbf{P}$)*:*

(*i*) *if it is* $\bot$*, then* $\mathbf{P}$ *is inconsistent;*
(*ii*) *otherwise, it is a* k*-hyperarc consistent CSP equivalent to* $\mathbf{P}$*.*

Proof. First we show that the subprocedure Project preserves equivalence, in the following sense. Let R' be the multiset of constraints before the jth invocation of Project in Line 5, let Y and i be the parameters of such invocation, and let R'' be the multiset of constraints computed by the jth execution of Lines 14-25. We aim to show that for every $t \in l(X)$,

$$\bigodot_{C_Y \in R'} C_Y(t|_Y) = \bigodot_{C_Y \in R''} C_Y(t|_Y), \tag{2}$$

that is, problems $(X, D, R', \mathbf{A})$ and $(X, D, R'', \mathbf{A})$ are equivalent.

Let $t \in l(X)$ and let $t|_{\{i\}} = a \in D_i$ such that $C_{\{i\}}(a) > \bot$ (Line 15). Clearly, $t|_{Y\setminus\{i\}} \in l(Y \setminus \{i\})$. In Line 16, x is settled to a maximal element in the poset

$$\{C_Y(t|_{Y\setminus\{i\}} \cdot t|_{\{i\}}) \mid t|_{Y\setminus\{i\}} \in l(Y \setminus \{i\})\},$$

so that by construction $C_Y(t|_Y) \leq x$. By Line 17, the constraint $C_{\{i\}}(t|_{\{i\}})$ in R' becomes

$$C_{\{i\}}(t|_{\{i\}}) \odot x$$

in R'', and by Line 22, at some iteration of the loop in Lines 21-23, the constraint $C_Y(t|_Y)$ in R' becomes

$$x \to C_Y(t|_Y)$$

in R''. Now, we claim that:

$$C_{\{i\}}(t|_{\{i\}}) \odot C_Y(t|_Y) = (C_{\{i\}}(t|_{\{i\}}) \odot x) \odot (x \to C_Y(t|_Y)).$$

Indeed, in light of Fact 2(iii) and the aforementioned fact that $C_Y(t|_Y) \leq x$,

$$\begin{aligned}(C_{\{i\}}(t|_{\{i\}}) \odot x) \odot (x \to C_Y(t|_Y)) &= C_{\{i\}}(t|_{\{i\}}) \odot (x \odot (x \to C_Y(t|_Y))) \\ &= C_{\{i\}}(t|_{\{i\}}) \odot (x \wedge C_Y(t|_Y)) \\ &= C_{\{i\}}(t|_{\{i\}}) \odot C_Y(t|_Y).\end{aligned}$$

Eventually, PROJECT does not modify constraints $C_Z \in R'$ such that $Z \neq \{i\}$ and $Z \neq Y$, so that,

$$\bigodot_{C_Z \in R', Z \neq \{i\}, Z \neq Y} C_Z(t|_Z) = \bigodot_{C_Z \in R'', Z \neq \{i\}, Z \neq Y} C_Z(t|_Z).$$

Thus, since $z' = z''$ implies $z \odot z' = z \odot z''$ in $\mathbf{A}$ for every $z, z', z'' \in A$ by Fact 1(ii), we conclude that (2) holds.

Now suppose that the algorithm outputs $\bot$ in Line 7. We claim that the input problem $\mathbf{P} = (X, D, P, \mathbf{A})$ is inconsistent. Indeed, let j be such that after the jth execution of PROJECT, say over parameters Y and i, it holds that $C_{\{i\}}(a) = \bot$ for each $a \in D_i$. Let P' be the multiset of constraints computed by such jth execution. Since PROJECT preserves equivalence, $\mathbf{P'} = (X, D, P', \mathbf{A})$ is equivalent to $\mathbf{P}$. But, by Proposition 3, $\mathbf{P'}$ is inconsistent, so that $\mathbf{P}$ is inconsistent too.

Next suppose that the algorithm outputs $\mathbf{P'} = (X, D, P', \mathbf{A})$ in Line 13. We claim that the output problem is k-hyperarc consistent and equivalent to the input problem $\mathbf{P} = (X, D, P, \mathbf{A})$. For equivalence, simply note that PROJECT preserves equivalence. For k-hyperarc consistency, first note that every $i \in [n]$ is such that $C_{\{i\}}(a) > \bot$ for some $a \in D_i$. Indeed, this holds in the input problem $\mathbf{P}$ without loss of generality, and each execution of PROJECT, which possibly pushes some $C_{\{i\}}(a)$ down to $\bot$ in Line 17, is followed by the check of Lines 18-20.

Now, let $Y \subseteq X$ be such that $2 \leq |Y| \leq k$, $i \in Y$ and $C_Y \in P'$, and let $a \in D_i$ be such that $C_{\{i\}}(a) > \bot$. We claim that there exists $t \in l(Y \setminus \{i\})$ such that

$$C_{\{i\}}(a) = C_{\{i\}}(a) \odot C_Y(t \cdot a).$$

Note that, by Fact 1(i), equality holds if $C_Y(t \cdot a) = \top$. Let R' and R'' be respectively the multisets of constraints before and after the last execution of PROJECT on input Y and i. Let $t \in l(Y \setminus \{i\})$ be such that $C_Y(t \cdot a)$ is the maximal element in $\{C_Y(t \cdot a) \mid t \in l(Y \setminus \{i\})\}$ assigned to x in Line 16. Thus, at some iteration of loop in Lines 21-23, we have that the constraint $C_Y(t \cdot a)$ in R' is updated to $x \to C_Y(t \cdot a)$ in R''. But, by Fact 2(i),

$$x \to C_Y(t \cdot a) = C_Y(t \cdot a) \to C_Y(t \cdot a) = \top,$$

therefore, $C_Y(t \cdot a) = \top$ in R''. Noticing that subsequent assignments to $C_Y(t \cdot a)$ during the main loop have the form $x \to \top$, which is equal to $\top$ by Fact 2(i), the claim is settled. □

Theorem 5. *Let* $\mathbf{P}$ *be a CSP, and let* $\mathbf{P'} = k$-HYPERARC-CONSISTENCY($\mathbf{P}$). *Then,* $\mathbf{P'}$ *is a* k*-hyperarc consistent CSP equivalent to* $\mathbf{P}$*, computed in polynomial time in the size of* $\mathbf{P}$.

Acknowledgments. The author thanks the anonymous referees for careful comments on the preliminar version of this paper.

References

1. Ansótegui, C., Bonet, M.L., Levy, J., Manyà, F.: The Logic Behind Weighted CSP. In: In: 20th International Joint Conference on Artificial Intelligence, pp. 32–37 (2007)
2. Bistarelli, S., Gadducci, F.: Enhancing Constraints Manipulation in Semiring-Based Formalisms. In: 17th European Conference on Artificial Intelligence, pp. 63–67. IOS Press (2006)
3. Bova, S., Montagna, F.: The Consequence Relation in the Logic of Commutative GBL-Algebras is PSPACE-complete. Theor. Comput. Sci. 410(12-13), 1143–1158 (2009)
4. Bistarelli, S., Montanari, U., Rossi, F.: Semiring-Based Constraint Satisfaction and Optimization. J. ACM 44(2), 201–236 (1997)
5. Bistarelli, S., Montanari, U., Rossi, F., Schiex, T., Verfaillie, G., Fargier, H.: Semiring-Based CSPs and Valued CSPs: Frameworks, Properties, and Comparison. Constraints 4(3), 199–240 (1999)
6. Cohen, D.A., Cooper, M.C., Jeavons, P.: An Algebraic Characterisation of Complexity for Valued Constraint. In: 12th International Conference on Principles and Practice of Constraint Programming, pp. 107–121. Springer (2006)
7. Cohen, D.A., Cooper, M.C., Jeavons, P., Krokhin, A.A.: The Complexity of Soft Constraint Satisfaction. Artif. Intell. 170, 983–1016 (2006)
8. Cooper, M.C., de Givry, S., Schiex, T.: Optimal Soft Arc Consistency. In: 20th International Joint Conference on Artificial Intelligence, pp. 68–73 (2007)
9. Cooper, M.C., de Givry, S., Schiex, T., Sánchez, M., Zytnicki, M.: Virtual Arc Consistency for Weighted CSP. In: 23rd AAAI Conference on Artificial Intelligence, pp. 253-258 (2008)
10. Cignoli, R., Esteva, F., Godo, L., Torrens, A.: Basic Fuzzy Logic is the Logic of Continuous t-Norms and their Residua. Soft Comput. 4(2), 106–112 (2000)
11. Cooper, M.C., Schiex, T.: Arc Consistency for Soft Constraints. Artif. Intell. 154(1-2), 199–227 (2004)
12. Dechter, R.: Constraint Processing. Morgan Kaufmann (2003)
13. Galatos, N., Jipsen, P., Kowalski, T., Ono, H.: Residuated Lattices: An Algebraic Glimpse at Substructural Logics. Elsevier (2007)
14. Gottwald, S.: A Treatise on Many-Valued Logics. Research Studies Press (2000)
15. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer (1998)
16. Jipsen, P., Montagna, F.: On the Structure of Generalized BL-Algebras. Algebra Univ. 55, 226–237 (2006)
17. Klement, E.P., Mesiar, R., Pap, E.: Triangular Norms. Kluwer (2000)
18. Larrosa, J., Schiex, T.: Solving Weighted CSP by Maintaining Arc Consistency. Artif. Intell. 159(1-2), 1–26 (2004)
19. Montanari, U.: Networks of Constraints: Fundamental Properties and Applications to Picture Processing. Inform. Sciences 7, 95–132 (1974)
20. Pavelka, J.: On Fuzzy Logic I, II, III. Zeitschr. f. Math. Logik und Grundlagen der Math. 25, 45–52,119–134,447–464 (1979)
21. Rasiowa, H.:An Algebraic Approach to Non-Classical Logics. North-Holland (1974)
22. Ward, M., Dilworth, R.P.: Residuated Lattices. T. Am. Math. Soc 45, 335–354 (1939)