Extending Soft Arc Consistency Algorithms to Non-Invertible Semirings, with an Application to Multi-Criteria Problems*

Stefano Bistarelli[1,2,3], Fabio Gadducci[4], Emma Rollon[5] and Francesco Santini[2,3]

[1] Dipartimento di Matematica e Informatica Università di Perugia, Italy
bista@dipmat.unipg.it
[2] Dipartimento di Scienze, Università "G. d'Annunzio" di Chieti-Pescara, Italy
bista@sci.unich.it, santini@sci.unich.it
[3] Istituto di Informatica e Telematica (IIT-CNR) Pisa, Italy
stefano.bistarelli@iit.cnr.it, francesco.santini@iit.cnr.it
[4] Dipartimento di Informatica, Università di Pisa, Italy
gadducci@di.unipi.it
[5] Department of Software, Technical University of Catalonia, Barcelona, Spain
erollon@lsi.upc.edu

Abstract. We extend arc consistency algorithms proposed in the literature in order to deal with not necessarily invertible semirings. As a result, consistency algorithms can now be used as a preprocessing procedure in soft CSPs defined over a larger class of semirings: either partially ordered, or with non idempotent $\times$, or not closed $\div$ operator, or constructed as cartesian product or Hoare power sets of any semiring (which can be used for multicriteria CSPs). To reach this objective, we first show that each semiring can be transformed into a new one where the $+$ is instantiated with the Least Common Divisor (LCD) between the elements of the semiring. The LCD value corresponds to the amount we can "safely move" from the binary constraint to the unary one in the consistency algorithm (when $\times$ is not idempotent). We then propose an arc consistency algorithm which takes advantage of this operator.

1 Introduction and Motivations

Constraint propagation embeds any reasoning which consists in explicitly forbidding values or combinations of values for some variables of a problem because a given subset of its constraints cannot be satisfied otherwise. A very important mean to accomplish this task is represented by local consistency algorithms.

Local consistency [1] is an essential component of a constraint solver: a local property with an enforcing algorithm, often polynomial time, that transforms a

* Research partially supported by the EU FP6-IST IP 16004 SENSORIA and by the Spanish Min. of Science and Innovation through the projects TIN2006-15387-C03-0
.

classical constraint network into an equivalent network that satisfies the property. If this equivalent network is empty, then the initial problem/network is obviously inconsistent, allowing to detect some inconsistencies very efficiently.

The idea of the semiring-based framework [1, 2] was to further extend the classical constraint notion, and to do it with a formalism that could encompass most of the existing extensions, with the aim to provide a single environment where properties could be proven once and for all, and inherited by all the instances. At the technical level, this was done by adding to the usual notion of *Constraint Satisfaction Problem* [1] (CSP) the concept of a structure representing the levels of satisfiability of the constraints. Such a structure is a set with two operations (see Sec. 2 for further details): one (written $+$) is used to generate an ordering over the levels, while the other one ($\times$) is used to define how two levels can be combined and which level is the result of such combination. Because of the properties required on such operations, this structure is similar to a semiring (see Sec. 2): from here the terminology of "semiring-based soft constraint" [2–4], that is, constraints with several levels of satisfiability, and whose levels are (totally or partially) ordered according to the semiring structure. In general, problems defined according to the semiring-based framework are called *Soft Constraint Satisfaction Problems* (soft CSPS or SCSPs).

In the literature, many local consistency algorithms have been proposed for soft CSPs: while classical consistency algorithms [5] aim at reducing the size of constraint problems, soft consistency algorithms work by explicating the error information that is originally implicit in the problem. The most recent ones exploit *invertible* semirings (see Sec. 2), providing, under suitable conditions, an operator $\div$ that is the inverse of $\times$, i.e., such that $(a \div b) \times b = a$ (see two alternative proposals in [6] and [7, 8], respectively).

In this paper we aim at generalizing the previous consistency algorithms to not necessarily invertible semirings. In particular, we first show how to distill from a semiring a novel one, such that its $+$ operator corresponds to the *Least Common Divisor* (*LCD*) operator (see Sec. 3) of the elements in the semiring preference set. We then show, and this represents the practical application of the first theoretical outcome, how to apply the derived semiring inside soft arc consistency algorithms in order to extend their use to not necessarily invertible semiring structures, thus leading to a further generalization of soft arc consistency techniques. In words, the value represented by the LCD corresponds to the amount we can "safely move" from the binary constraint to the unary one in the consistency algorithm. Summing up, this paper extends the use of consistency algorithms beyond the limits imposed by the proposals in [6, 7]. As a practical and very important example, the new consistency algorithms can be applied to multicriteria problems, where the Hoare Powerdomain of the Cartesian product of multiple semirings represents the set of partially ordered solutions [9].

The paper is organized as follows: Sec. 2 summarizes the background notions about semirings and soft constraints. Sec. 3 shows how to assemble the new LCD operator by transforming a semiring, while Sec. 4 proposes its use inside a local consistency algorithm for soft CSPs. Then, Sec. 5 shows the algorithm execution over a multicriteria problem. The final remarks are provided in Sec. 6.

2 Preliminaries

Semirings provide an algebraic framework for the specification of a general class of combinatorial optimization problems. Outcomes associated to variable instantiations are modeled as elements of a set A, equipped with a sum and a product operator. These operators are used for combining constraints: the intuition is that the sum operator induces a partial order $a \leq b$, meaning that b is a better outcome than a; whilst the product operator denotes the aggregation of outcomes coming from different soft constraints.

2.1 The Algebra of Semirings

This section reviews the main concepts, adopting the terminology used in [3, 6]. A (commutative) semiring is a five-tuple $\mathcal{K} = \langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$ such that A is a set, $\mathbf{1}, \mathbf{0} \in A$, and $+, \times : A \times A \to A$ are binary operators making the triples $\langle A, +, \mathbf{0} \rangle$ and $\langle A, \times, \mathbf{1} \rangle$ commutative monoids (semigroups with identity), satisfying distributivity ($\forall a, b, c \in A. a \times (b + c) = (a \times b) + (a \times c)$) and with $\mathbf{0}$ as annihilator element for $\times$ ($\forall a \in A. a \times \mathbf{0} = \mathbf{0}$). A semiring is *absorptive* if additionally $\mathbf{1}$ is an annihilator element for $+$ ($\forall a \in A. a + \mathbf{1} = \mathbf{1}$). [1]

Let $\mathcal{K}$ be an absorptive semiring. Then, the operator $+$ of $\mathcal{K}$ is idempotent. As a consequence, the relation $\langle A, \leq \rangle$ defined as $a \leq b$ if $a + b = b$ is a partial order and, moreover, $\mathbf{1}$ is its top element. If additionally $\mathcal{K}$ is also *idempotent* (that is, the product operator $\times$ is idempotent), then the partial order is actually a *lattice*, since $a \times b$ corresponds to the greatest lower bound of a and b. For the rest of the paper, we fix the absorbing semiring $\mathcal{K} = \langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$.

An absorptive semiring is *invertible* if whenever $a \leq b$, there exists an element $c \in A$ such that $b \times c = a$ or, in other words, if the set $Div(a, b) = \{c \mid b \times c = a\}$ is not empty. It is *uniquely* invertible if that set is actually a singleton whenever $b \neq \mathbf{0}$. All classical soft constraint instances (i.e. *Classical CSPs*, *Fuzzy CSPs*, *Probabilistic CSPs* and *Weighted CSPs*) are indeed invertible, and uniquely so.

A *division* operator $\div$ is a kind of weak inverse of $\times$, such that $a \div b$ returns an element chosen from $Div(a, b)$. There are currently two alternatives in the literature: the choice in [6] favours the maximum of such elements (whenever it exists), the choice in [7] favours the minimum (whenever it exists). While the latter is better computationally, the former encompasses more semiring instances. [2] All of our results in the following sections can be applied for any choice of $\div$.

[1] The absorptiveness property is equivalent to $\forall a, b \in A. a + (a \times b) = a$, that is, any element $a \times b$ is actually "absorbed" by either a or b.

[2] For example, *complete* semirings, i.e., those which are closed wrt. infinite sums, and the distributivity law holds also for an infinite number of summands. See [6] for a throughout comparison between the two alternatives.

Example 1. Let us consider the *weighted* semiring $\mathcal{K}_w = \langle \mathbb{N} \cup \{\infty\}, \min, +, 0, \infty \rangle$. The $+$ and $\times$ operators are min and $+$ with their usual meaning over the naturals. The ∞ value is handled in the usual way (i.e. $\min\{\infty, a\} = a$ and $\infty + a = a$ for all a). This semiring is widely used to model and solve a variety of combinatorial optimization problems [10]. Note that the induced order is total and corresponds to the inverse of the usual order among naturals (e.g. $9 \leq 6$ because $\min\{9, 6\} = 6$). Note as well that $\mathcal{K}_w$ is uniquely invertible. The division corresponds to the usual substraction over the naturals (e.g. $a \div b = a - b$). The only case that deserves special attention is $\infty \div \infty$, because any value of the semiring satisfies the division condition. In [6] and [7] it is defined as 0 and ∞, respectively.

Consider now semiring $\mathcal{K}_b = \langle \mathbb{N}^+ \cup \{\infty\}, min, \times, \infty, 1 \rangle$, where $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$. The $+$ and $\times$ operators are now min, $\times$ with the usual meaning over the naturals. This semiring is a slightly modified version of a semiring proposed in [11] in order to deal with bipolar preferences. As before, the semiring is totally ordered. However, it is not invertible: for example, even if $9 \leq 6$, clearly there is no $a \in \mathbb{N}^+ \cup \{\infty\}$ such that $6 \times a = 9$. Intuitively, $\mathbb{N}^+ \cup \{\infty\}$ is not closed under the arithmetic division, the obvious inverse of the arithmetic multiplication.

2.2 Soft Constraints Based on Semirings

The aim of this section is to briefly recall the main concepts of the semiring-based approach to the soft CSPs framework.

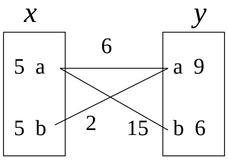
Definition 1 (constraints). *Let $\mathcal{K} = \langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$ be an absorptive semiring; let V be an ordered set of variables; and let D be a finite domain of interpretation for V. Then, a* constraint *$(V \to D) \to A$ is a function associating a value in A to each assignment $\eta : V \to D$ of the variables.*

We then define $C = \eta \to A$ as the set of all constraints that can be built starting from $\mathcal{K}$, V and D. The application of a constraint function $c : (V \to D) \to A$ to a mapping $\eta : V \to D$, is noted $c\eta$. Note that even if a constraint involves all the variables in V, it can depend on the assignment of a finite subset of them, called its support. For instance, a binary constraint c with $supp(c) = \{x, y\}$ is a function $c : (V \to D) \to A$ which depends only on the assignment of variables $\{x, y\} \subseteq V$. The support corresponds to the classical notion of scope of a constraint. We often refer to a constraint with support I as c_I. Moreover, an assignment over a support I of size k is concisely represented by a tuple t in D^k and we often write $c_I(t)$ instead of $c_I\eta$.

We now present the extension of the basic operations (namely, combination, division and projection) to soft constraints.

Definition 2 (combination and division). *The* combination *operator $\otimes : C \times C \to C$ is defined as $(c_1 \otimes c_2)\eta = c_1\eta \times c_2\eta$ for any two constraints c_1, c_2.*

The division *operator $\oplus\!\!\!\!\div : C \times C \to C$ is defined as $(c_1 \oplus\!\!\!\!\div c_2)\eta = c_1\eta \div c_2\eta$ for any two constraints c_1, c_2 such that $c_1 \sqsubseteq c_2$ (i.e., such that $c_1\eta \leq c_2\eta$ for all η).*



a)

x	y	
a	a	20
a	b	26
b	a	16
b	b	11

b)

x	y	
a	a	270
a	b	450
b	a	90
b	b	30

c)

Fig. 1. A soft CSP and the combination of its constraints wrt. semirings $\mathcal{K}_w$ and $\mathcal{K}_b$.

Informally, performing the $\otimes$ or the $\oplus$ between two constraints means building a new constraint whose support involves all the variables of the original ones, and which associates with each tuple of domain values for such variables a semiring element which is obtained by multiplying or, respectively, dividing the elements associated by the original constraints to the appropriate sub-tuples.

Definition 3 (projection). *Let $c \in C$ be a constraint and $v \in V$ a variable. The* projection *of c over $V - \{v\}$ is the constraint c' such that $c'\eta = \sum_{d \in D} c\eta[v := d]$.*

We denote such projection as $c \Downarrow_{(V-\{v\})}$. The projection operator is inductively extended to a set of variables $I \subseteq V$ by $c \Downarrow_{(V-I)} = c \Downarrow_{(V-\{v\})} \Downarrow_{(V-\{I-\{v\}\})}$. Informally, projecting means eliminating variables from the support.

Definition 4 (soft CSPs). *A* soft constraint satisfaction problem *is a pair $\langle C, Y\rangle$, where C is a set of constraints and $Y \subseteq V$.*

The set Y contains the variables of interest for the constraint set C.

Definition 5 (solutions). *The* solution *of a soft CSP $P = \langle C, Y\rangle$ is defined as the constraint $Sol(P) = (\bigotimes C) \Downarrow_Y$.*

The solution of a soft CSP is obtained by combining all constraints, and then projecting over the variables in Y. In this way we get the constraint with support (not greater than) Y which is "induced" by the entire soft CSP.

A tightly related notion, one wehich is quite important in combinatorial optimization problems, is the best level of consistency.

Definition 6. *The* best level of consistency *of a soft CSP problem $P = \langle C, Y\rangle$ is defined as the constraint $blevel(P) = (\bigotimes C) \Downarrow_\emptyset$.*

Example 2. Figure 1.a) shows a soft CSP with variables $V = \{x, y\}$ and values $D = \{a, b\}$. For the purpose of the example, all the variable in V are of interest. The problem has two unary soft constraints c_x, c_y and one binary constraint c_{xy}. Each rectangle represents a variable and contains its domain values. Besides each domain value there is a semiring value given by the corresponding unary constraint (for instance, c_y gives value 9 to any labeling in which variable y takes value a). Binary constraints are represented by labeled links between pairs of values. For instance, c_{xy} gives value 15 to the labeling in which variable x and y take values a and b. If a link is missing, its value is the unit $\mathbf{1}$ of the semiring.

Figure 1.b) shows the combination of all constrains (i.e. $c_x \otimes c_y \otimes c_{xy}$) assuming the weighted semiring K_w. Each table entry is the sum of the three corresponding semiring values. In this case, the best level of solution is 11 the minimum over all the entries. Figure 1.c) shows the combination of all constrains assuming the bipolar semiring K_b. It is different wrt. the previous case, because semiring values are now multiplied. As before, the best level of solution is the minimum over all the entries which, in this case, is 30.

2.3 Local Consistency

Soft local consistencies are properties over soft CSPs that a given instance may or may not satisfy. For the sake of simplicity, in this paper we restrict ourselves to the simplest consistencies. However, all the notions and ideas presented in subsequent sections can be easily generalized to more sophisticated ones.

In the sequel, we assume that soft CSPs are binary (i.e., no constraint has arity greater than 2). We also assume the existence of a unary constraint c_x for every variable x, and of a zero-arity constraint (i.e. a constant), noted $c_{\mathbf{0}}$: if such constraints are not defined, we consider dummy ones: $c_x(a) = \mathbf{1}$ for all $a \in D$ and $c_{\mathbf{0}} = \mathbf{1}$. One effect of local consistency is to detect and remove unfeasible values. For that purpose we define the current domain variable x as $D_x \subseteq D$.

Definition 7. *Let $P = \langle C, Y \rangle$ be a binary soft CSP.*

- *Node consistency (NC). Value $a \in D_x$ is NC if $c_{\mathbf{0}} \times c_x(a) > \mathbf{0}$. Variable x is NC if $\sum_{a \in D_x} c_x(a) = \mathbf{1}$. P is NC if every variable is NC.*
- *Arc consistency (AC). Value $a \in D_x$ is AC wrt. $c_{x,y}$ if $\sum_{b \in D_y} c_{x,y}(a,b) = \mathbf{1}$. Variable x is AC if all its values are AC wrt. every binary constraint such that x is in its support. P is AC if every variable is AC and NC.*

Each property should come with an enforcing algorithm that transforms the problem into an equivalent one that satisfies the property. Enforcing algorithms are based on the concept of local consistency rule.

Definition 8 (local consistency rule [6]). *Let c' and c be two constraints such that $supp(c') \subset supp(c)$ and let $Y = supp(c) \setminus supp(c')$. A* local consistency rule *involving c' and c, denoted $CR(c', c)$, consists of two steps*

- *Aggregate to c' information induced by c*

$$c' := c' \otimes (c \Downarrow_Y)$$

- *Compensate in c the information aggregated to c' in the previous step*

$$c := c \oplus\!\!\!\!\div (c \Downarrow_x).$$

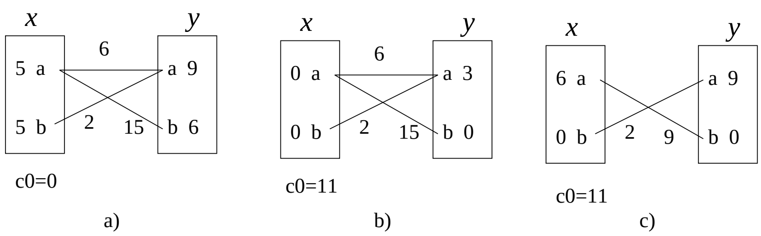


Fig. 2. Three equivalent soft CSP instances (the semiring being $\mathcal{K}_w$). The first one is arc and node inconsistent. The second one is node consistent and arc inconsistent. The third one is node and arc consistent.

It is important to note that local consistency rules are only defined for soft CSPs with invertible semirings, because $\div$ is only defined for them. The fundamental property of the above local consistency rule is that it does not change the solution of soft CSPs defined on invertible semirings [6].

Example 3. Figure 2.a) shows the soft CSP we have chosen as our running example. Assuming the $\mathcal{K}_w$ semiring, the problem is not NC. It can be made NC by applying the local consistency rule twice, to $c_{\mathbf{0}}, c_x$ and $c_{\mathbf{0}}, c_y$, which increases $c_{\mathbf{0}}$ and decreases the unary constraints. The resulting equivalent soft CSP is depicted in Figure 2.b). However, it is not AC. It can be made arc consistent by applying the local consistency rule to $c_x, c_{x,y}$. The resulting equivalent soft CSP is depicted in Figure 2.c). This problem is more explicit than the original one. In particular, the zero-arity constraint $c_{\mathbf{0}}$ contains the best level of consistency.

3 Defining an LCD-based Semiring Transformation

Many absorptive semirings are not invertible: see e.g. [6] for some references. Nevertheless, we would like to apply some consistency rules also to these cases.

The aim of the section is to show how to distill from a semiring $\mathcal{K}$ a new semiring $LCD_{\preceq}(\mathcal{K})$ such that the latter is invertible. In this section we show the construction of the structure $LCD_{\preceq}(\mathcal{K})$, prove it to be a semiring, and state a conservativity result, namely, that $\mathcal{K}$ and $LCD_{\preceq}(\mathcal{K})$ coincide, if $\mathcal{K}$ is invertible.

Technically, we exploit a notion of *least common divisor*: for any two elements of a semiring, we consider the set of common divisors, and we assume the existence of at least one minimal element among such divisors: intuitively, these are the "worst" elements according to the ordering on the semiring.

Definition 9 (common divisor). *Let $\mathcal{K}$ be an absorptive semiring and let $a, b \in A$. The* divisors *of a is the set $Div(a) = \{c \mid \exists d. c \times d = a\}$; the* common divisors *of a and b is the set $CD(a, b) = Div(a) \cap Div(b)$.*

Clearly, the set $CD(a, b)$ is never empty, since it contains at least $\mathbf{1}$. Moreover, its elements form a partial order, from which we may identify the minimal ones.

Definition 10 (least common divisor). *Let $\mathcal{K}$ be an absorptive semiring and $a, b \in A$. The* least common divisors *of a, b are the minimal of the set $CD(a, b)$, i.e., the elements of the set $LCD(a, b) = \{c \in CD(a, b) \mid \nexists d \in CD(a, b).d < c\}$.*

We say that $\mathcal{K}$ admits bound least common divisors is the set $LCD(a, b)$ is finite and not empty for any $a, b \in A$.

We can characterize the choice of a least common divisor if a linearization of the partial order associated to the semiring is found.

Definition 11 (linearized least common divisor). *Let $\mathcal{K}$ be an absorptive semiring. A* linearization *of the partial order $\leq_K$ associated to $\mathcal{K}$ is a total order $\preceq_K$ which is compatible with $\leq_K$, i.e., such that $\forall a, b \in A.a \leq_K b \implies a \preceq_K b$.*

If $\mathcal{K}$ has bound least common divisors, $LCD_{\preceq_K}(a, b)$ denotes the minimum of the set $LCD(a, b)$ according to $\preceq_K$ for any $a, b \in A$.

In other words, the $LCD_{\preceq}(a, b)$ operator returns a single element of $LCD(a, b)$, according to a given ordering. In the following, for any finite set $E \subseteq A$, $LCD(E)$ and $LCD_{\preceq}(E)$ denote the obvious associative extensions.

We can finally prove the main result of this section, i.e., the existence of a semiring whose sum operator is based on $LCD_{\preceq}$. This new semiring is used by the consistency algorithm in Sec. 4.

Theorem 1 (LCD-based semiring). *Let $\mathcal{K}$ be an absorptive semiring with bound least common divisors, and let $\preceq$ be one of its linearizations. Then, the tuple $LCD_{\preceq}(\mathcal{K}) = \langle A, LCD_{\preceq}, \times, \mathbf{0}, \mathbf{1} \rangle$ is an absorptive and invertible semiring.*

Proof. As for proving that $LCD_{\preceq}(\mathcal{K})$ is an absoprtive semiring, it just suffices to check out each single property, noting that they hold since some choices are forced by the linearization. As an example, $LCD(a, \mathbf{0})$ coincides with $Div(a)$, and the linearization tells us that $LCD_{\preceq}(a, \mathbf{0}) = a$. Similar considerations hold for associativity and distributivity. As for invertibility, by definition $a \leq_{LCD_{\preceq}} b$ implies that $LCD_{\preceq}(a, b) = b$. □

As a final result, we need to check out what is the outcome of the application of the $LCD_{\preceq}$ construction to an already invertible semiring.

Proposition 1 (conservativity). *Let $\mathcal{K}$ be an aborptive and invertible semiring. Then, $\mathcal{K}$ and $LCD_{\preceq_K}(\mathcal{K})$ are the same semiring for any linearization $\preceq_K$.*

The proof is immediate: note that if $\mathcal{K}$ is invertible, then by definition $a + b \in CD(a, b)$, and the element surely is the greatest lower bound of the set.

Example 4. Recall the non invertible semiring $\mathcal{K}_b = \langle \mathbb{N}^+ \cup \{\infty\}, min, \times, \infty, 1 \rangle$. Since this semiring is totally ordered, we ignore linearizations. By definition, the least common divisor of a and b is $LCD(a, b) = \max\{c \mid\ \exists d.c \times d = a,\ \exists e.c \times e = b\}$ which corresponds with the arithmetic notion of greatest common divisor. The LCD transformation of $\mathcal{K}_b$ is $LCD(\mathcal{K}_b) = \langle \mathbb{N}^+ \cup \{\infty\}, LCD(a, b), \times, \infty, 1 \rangle$. The partial order induced by $LCD(\mathcal{K}_b)$ is $a \leq b$ if $LCD(a, b) = b$, i.e., if b is a divisor of a. Finally, the division operator of $LCD(\mathcal{K}_b)$ is the usual arithmetic division. However, it is closed because it is restricted to divisible pairs of numbers.

4 LCD-based Local Consistency

This section generalizes local consistencies to soft CSPs with non invertible semirings. The leading intuition is to replace the + operator by the $LCD_{\preceq}$ operator also in the original definition of the local consistency rule. The value represented by the found LCD corresponds to the amount we can "safely move" from the binary constraint to the unary one in order to enforce consistency. This very intuitive idea can be applied also over non invertible semirings.

Definition 12. *Let $P = \langle C, Y \rangle$ be a binary soft CSP defined over semiring $\mathcal{K}$, and let $LCD_{\preceq}$ be alinearized least common divisor operator.*

- *LCD Node consistency (LCD-NC). Value $a \in D_x$ is LCD-NC if $c_{\mathbf{0}} \times c_x(a) > \mathbf{0}$. Variable x is LCD-NC if $LCD(\{c_x(a) \mid a \in D_x\}) = \mathbf{1}$. P is LCD-NC if every variable is LCD-NC.*
- *LCD Arc consistency (LCD-AC). Value $a \in D_x$ is LCD-AC wrt. $c_{x,y}$ if $LCD(\{c_{x,y}(a, b) \mid b \in D_y\}) = \mathbf{1}$. Variable x is LCD-AC if all its values are LCD-AC wrt. every binary constraint such that x is in its support. P is LCD-AC if every variable is LCD-AC and LCD-NC.*

Definition 13 (LCD local consistency rule). *Let $P = \langle C, Y \rangle$ be a soft CSP defined over semiring $\mathcal{K}$. A* LCD local consistency rule *involving c' and c, noted $LCD\text{-}CR(c', c)$, is like a classical local consistency rule where all operations (i.e, combination, projection and division) are done using the $LCD(\mathcal{K})$ semiring.*

Theorem 2. *Let P be a soft CSP. The application of the LCD local consistency rules of Definition 13 does not change its solution.*

Example 5. Figure 3.a) shows our running example. Assuming the $\mathcal{K}_b$ semiring, the problem is not LCD-NC. It can be made LCD-NC by applying the LCD consistency rule twice, to $c_{\mathbf{0}}, c_x$ and $c_{\mathbf{0}}, c_y$. The resulting equivalent soft CSP is depicted in Figure 2.b). Note that it is LCD-NC but not NC. It is not LCD-AC and can be made so by applying the LCD consistency rule twice, first to $c_x, c_{x,y}$ (producing Figure 3.c)) and then to $c_y, c_{x,y}$ (producing Figure 3.d). The resulting problem is not LCD-NC (due to variable y). It can be made LCD-NC and LCD-AC by applying the LCD consistency rule to $c_{\mathbf{0}}, c_y$, resulting the problem in Figure 3.e). This problem is more explicit than the original one. In particular, the zero-arity constraint $c_{\mathbf{0}}$ contains the best level of consistency.

The previous LCD node and arc consistency properties can be enforced by applying appropriated LCD consistency rules. Figure 4 shows a LCD-AC enforcing algorithm. It is based on the AC algorithm of [10]. For simplicity, we assume that no empty domain is produced. It uses two auxiliary functions: $PruneVar(x)$ prunes not LCD-NC values in D_x and returns *true* if the domain is changed; $LCD\text{-}CR(c', c)$ iteratively applies the LCD consistency rule to c' and c until reaching a fixed point. Note that if the original semiring $\mathcal{K}$ is invertible, $LCD\text{-}CR(c', c)$ iterate only once. The main procedure AC() uses a queue Q containing those variables that may not be LCD-AC. Q should be initialized with all the variables to check at least once their consistency.

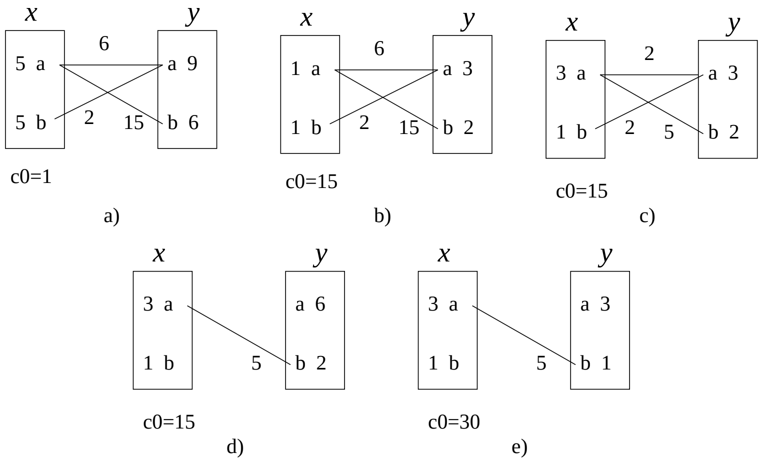


Fig. 3. Five equivalent soft CSP instances (the semiring being $\mathcal{K}_b$).The first one is LCD arc and node inconsistent. The second one is LCD node consistent and LCD arc inconsistent. The last one is LCD node and arc consistent.

5 Dealing with Multi-Objective Optimization

A *multi-criteria CSP* (MC-CSP) is a soft CSP composed by a family of p soft CSPs $\{P_i\}_{i=1}^p$. Each soft CSP P_i, which is called a criterion, is defined over a semiring $\mathcal{K}_i = \langle A_i, +_i, \times_i, \mathbf{0}_i, \mathbf{1}_i \rangle$. As shown in [9], a MC-CSP problem is defined over a semiring $\mathcal{K}_{mo}$ obtained via the so-called Hoare powerset of the cartesian product of the components. Before defining $\mathcal{K}_{mo}$ we review some basic concepts.

Let $\boldsymbol{A} = A_1 \times \ldots \times A_p$ be the set of all vectors with p components. Given two values $\boldsymbol{a}, \boldsymbol{b} \in \boldsymbol{A}$, let $\times$ and $+$ be the pointwise combination and addition of its elements (i.e., $\boldsymbol{a} \times \boldsymbol{b} = (a_1 \times_1 b_1, \ldots, a_p \times_p b_p)$ and $\boldsymbol{a} + \boldsymbol{b} = (a_1 +_1 b_1, \ldots, a_p +_p b_p)$), respectively. The comparison among vectors, denoted by $\leq$, is defined as $\boldsymbol{a} \leq \boldsymbol{b}$ iff $\boldsymbol{a} + \boldsymbol{b} = \boldsymbol{b}$. If $\boldsymbol{a} \leq \boldsymbol{b}$, we say that $\boldsymbol{b}$ is *better* of *dominates* $\boldsymbol{a}$.

Finally, et S be a set of vectors: we define its set of *non-dominated elements* $\|S\|$ as $\{\boldsymbol{u} \in S \mid \nexists \boldsymbol{v} \in S,\ \boldsymbol{u} < \boldsymbol{v}\}$

Lemma 1. *Let* $\{\langle A_i, +_i, \times_i, \mathbf{0}_i, \mathbf{1}_i \rangle\}_{i=1}^p$ *be a family of semirings. Then, also the structure* $\mathcal{K}_{mo} = \langle A_{mo}, +_{mo}, \times_{mo}, \mathbf{0}_{mo}, \mathbf{1}_{mo} \rangle$ *is a semiring, for*

- $A_{mo} = \{S \in \boldsymbol{A} \mid S = \|S\|\}$
- $S \times_{mo} T = \|\{\boldsymbol{u} \times \boldsymbol{v} \mid \boldsymbol{u} \in S,\ \boldsymbol{v} \in T\}\|$
- $S +_{mo} T = \|\{S \cup T\}\|$
- $\mathbf{0}_{mo} = \{(\mathbf{0}_1, \ldots, \mathbf{0}_p)\}$
- $\mathbf{1}_{mo} = \{(\mathbf{1}_1, \ldots, \mathbf{1}_p)\}$

If each semiring of the family is absortpive, then also $\mathcal{K}_{mo}$ *is so.*

The best level of consistency of a soft CSP problem P induced by semiring $\mathcal{K}_{mo}$ is the maximal set of non-dominated vectors, each one being the valuation

```
function PruneVar(x)
  change := false;
  for each a ∈ D_x do
      if c_x(a) × c_0 = 0 then
          D_x := D_x − {a};
          change := true;
  return change;
procedure LCD-AC(P, V)
  Q := V;
  while (Q ≠ ∅) do
      y := pop(Q);
      for each c_{x,y} ∈ C do LCD-CR(c_x, c_{xy});
      for each x ∈ V do LCD-CR(c_0, c_x);
      for each x ∈ V do
          if PruneVar(x) then Q := Q ∪ {x};
```

Fig. 4. Algorithm LCD-AC.

associated with one complete assignment. In the multi-objective literature, this set is called *efficient frontier* of P, denoted as $\mathcal{E}(P)$.

In general, the semiring $\mathcal{K}_{mo}$ is not invertible, even if If each semiring of the family is so. However, thanks to the LCD transformation, we can apply the local consistencies described in the previous section as follows.

Consider a multi-criteria soft CSP $P = \langle C, con \rangle$ composed by two soft CSPs, both defined on the weighted semiring $\mathcal{K}_w$. For our purposes, $con = V = \{x, y\}$ with domain values $D_x = \{a, b, c\}$ and $D_y = \{a, b\}$. The set of constraints C is composed by two unary constraints $c_x(x)$ and $c_y(y)$, and one binary constraint $c_{xy}(x, y)$. We can express these constraints extensionally with the tables below

x	
a	$\{(2,1)\} + \{(0,2),(1,0)\} + \{(0,1),(1,0)\}$
b	$\{(0,2),(1,0)\}$
c	$\{(2,1)\} + \{(0,2),(1,0)\}$

y	
a	$\{(0,0)\}$
b	$\{(0,0)\}$

x y	
a a	$\{(0,0)\}$
a b	$\{(0,0)\}$
b a	$\{(2,1)\} + \{(0,2),(3,0)\}$
b b	$\{(2,1)\} + \{(3,0)\}$
c a	$\{(2,1)\} + \{(0,2),(1,0)\}$
c b	$\{(2,1)\} + \{(1,1)\} + \{(0,2),(1,0)\} + \{(0,6),(2,0)\}$

where the valuation associated to each possible assignment of its variables is expressed as a decomposition of divisors that cannot be further decomposed. Moreover, we have the constraint $c_{\mathbf{0}} = \{(0,0)\}$.

First, let us see if the problem is node consistent (NC). The only variable that may not be NC is x. Operator $LCD(c_x(a), c_x(b), c_x(c))$ returns $\{\{(0,2),(1,0)\}\}$, which is the only common divisor among $c_x(a)$, $c_x(b)$ and $c_x(c)$. To make variable x NC, this valuation must be combined with the current $c_{\mathbf{0}}$ and divided from $c_x(a)$, $c_x(b)$ and $c_x(c)$. The unary constraint c_x is as follows,

x	
a	$\{(2,1)\} + \{(0,1),(1,0)\}$
b	$\{(0,0)\}$
c	$\{(2,1)\}$

and constraint $c_{\mathbf{0}} = \{(0,0)\} + \{(0,2),(1,0)\} = \{(0,2),(1,0)\}$.

Let us see if the problem is arc consistent. We have to verify that both variables x and y are AC. Consider variable x. Domain value $a \in D_x$ is AC because $LCD(c_{xy}(a,a), c_{xy}(a,b)) = \{\{(0,0)\}\}$. Domain value $b \in D_x$ is not AC because $LCD(c_{xy}(b,a), c_{xy}(b,b)) = \{\{(2,1)\}\}$. According to the AC enforcement algorithm, we combine divisor $\{(2,1)\}$ with $c_x(b)$ and divide $c_{xy}(ba)$ and $c_{xy}(bb)$ by it. Similarly, domain value $c \in D_x$ is not AC because $LCD(c_{xy}(c,a), c_{xy}(c,b)) = \{\{(2,1)\} + \{(0,2),(1,0)\}\}$. Again, it may be added to $c_x(c)$ and divided from $c_{xy}(ca)$ and $c_{xy}(cb)$. After these changes, the constraint tables are

x	
a	$\{(2,1)\} + \{(0,1),(1,0)\}$
b	$\{(2,1)\}$
c	$\{(2,1)\}$

x y	
a a	$\{(0,0)\}$
a b	$\{(0,0)\}$
b a	$\{(0,2),(3,0)\}$
b b	$\{(3,0)\}$
c a	$\{(0,0)\}$
c b	$\{(1,1)\} + \{(0,6),(2,0)\}$

Since we have modified the valuations in c_x, we have to revise if variable x is still NC. It is easy to see that there exists one common divisor among $c_x(a)$, $c_x(b)$ and $c_x(c)$, that is, $\{(2,1)\}$. Then, we must divide c_x by $\{(2,1)\}$ and combine this valuation with the current $c_{\mathbf{0}}$. Now, constraint c_x is

x	
a	$\{(0,1),(1,0)\}$
b	$\{(0,0)\}$
c	$\{(0,0)\}$

The current $c_\mathbf{0}$ is $\{(2,1)\} + \{(0,2),(1,0)\} = \{(2,3),(3,1)\}$.

Now, consider variable y: $LCD(c_{xy}(aa), c_{xy}(ba), c_{xy}(ca)) = \{\{(0,0)\}\}$, which means that no common divisor exists, except for trivial divisor $\{(0,0)\}$. As a consequence, variable y is already AC. Since both the variables in the problem are AC, also the problem is so.

Finally, consider the variable ordering $\{y, x\}$. Since the valuation of each of the domain values of variable y in c_y is $\{(0,0)\}$, the problem is also directional arc consistent (DAC) wrt. that order. It can be verified that also with the other direction, that is, wrt. variable ordering $\{x, y\}$, the problem is still DAC.

6 Conclusions and Further Works

We presented a technique for transforming any semiring into a novel one, whose sum operator corresponds to the LCD of the set of preferences. This new semiring can be cast inside local consistency algorithms and allows to extend their use to problems dealing with not invertible semirings. A noticeable application case is represented by multicriteria soft CSPs, where the (Hoare Powerdomain of the) Cartesian product of semirings represents the set of partially ordered solutions.

In the future we plan to extend existent libraries on crisp constraints in order to deal also with propagation for soft constraints, as proposed in this paper. Moreover, we would like to develop approximated algorithms following the ideas in [12], and focusing on multi-criteria problems.

References

1. Rossi, F., van Beek, P., Walsh, T.: Handbook of Constraint Programming. Foundations of Artificial Intelligence. Elsevier (2006)
2. Bistarelli, S., Montanari, U., Rossi, F.: Semiring-based constraint satisfaction and optimization. Journal of ACM **44**(2) (1997) 201–236
3. Bistarelli, S.: Semirings for Soft Constraint Solving and Programming. Volume 2962 of LNCS. Springer (2004)
4. Bistarelli, S., Rossi, F.: Semiring-based soft constraints. In Degano, P., Nicola, R.D., Meseguer, J., eds.: Concurrency, Graphs and Models. Volume 5065 of LNCS., Springer (2008) 155–173
5. Mackworth, A.K.: Consistency in networks of relations. Artificial Intelligence **8**(1) (1977) 99–118
6. Bistarelli, S., Gadducci, F.: Enhancing constraints manipulation in semiring-based formalisms. In Brewka, G., Coradeschi, S., Perini, A., Traverso, P., eds.: ECAI'06. Volume 141 of Frontiers in Artif. Intell. and Applications., IOS Press (2006) 63–67
7. Cooper, M., Schiex, T.: Arc consistency for soft constraints. Artificial Intelligence **154**(1–2) (2004) 199–227
8. Cooper, M.: High-order consistency in valued constraint satisfaction. Constraints **10**(3) (2005) 283–305
9. Bistarelli, S., Gadducci, F., Larrosa, J., Rollon, E.: A soft approach to multi-objective optimization. In de la Banda, M.G., Pontelli, E., eds.: ICLP'08. Volume 5366 of LNCS., Springer (2008) 764–768

10. Larrosa, J., Schiex, T.: Solving weighted CSP by maintaining arc consistency. Artificial Intelligence **159**(1-2) (2004) 1–26
11. Bistarelli, S., Pini, M.S., Rossi, F., Venable, K.B.: Bipolar preference problems: Framework, properties and solving techniques. In Azevedo, F., Barahona, P., Fages, F., Rossi, F., eds.: CSCLP'06. Volume 4651 of LNCS., Springer (2007) 78–92
12. Larrosa, J., Meseguer, P.: Exploiting the use of DAC in MAX-CSP. In Freuder, E.C., ed.: CP'96. Volume 1118 of LNCS., Springer (1996) 308–322