HAL open science

Propagation Properties of Min-closed CSPs

Guy Alain Narboni

To cite this version:

Guy Alain Narboni. Propagation Properties of Min-closed CSPs. International Conference on Logic Programming, 2014, Austria. pp.Vienna Summer of Logic 2014. hal-01015663

HAL Id: hal-01015663

https://hal.science/hal-01015663v1

Submitted on 27 Jun 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Propagation Properties of Min-closed CSPs

Guy Alain Narboni
Implexe, France
(*e-mail:* r-d@implexe.fr)

submitted 14 February 2014; revised 15 May; accepted 26 March as ICLP'14 Technical Communication

Abstract

Min-closed constraints are numerical relationships characterised by a simple property. Yet, with finite-domain variables, min-closed systems give rise to a polynomial class of Constraint Satisfaction Problems. Propagation alone checks them for satisfiability. Solving is therefore search-free. Can this result be generalized from a discrete to a continuous (or mixed) setting? In this paper, we investigate the use of interval solvers for handling constraints with real variables. We show that the completeness result observed in the discrete case gracefully degrades into a 'close approximation' property in the continuous case. When switching from finite to infinite domains, the pruning power of propagation remains intact in the sense that it provides a box enclosure whose lower bound cannot be further updated (even by domain splitting). Applications of this analysis to scheduling, rule-based reasoning and scientific simulation are briefly mentioned.

KEYWORDS: Interval solvers, Bounds-propagation, Min-closed (resp. Max-closed) constraints

1 Introduction

A constraint satisfaction problem (CSP) is a formal problem statement which involves a finite set of variables (or unknowns), together with their associated definition domains, and a finite conjunction of constraints (or requirements), interrelating the variables. Typically, in combinatorial problems, variables have a finite domain, represented as an interval over the integers. A binary variable for instance has a $\{0, 1\}$ domain, where 0 conventionally stands for false and 1 for true.

A numeric CSP can be viewed as a CSP in which the integrality constraint on a variable x is removed and replaced — if implicit — by an explicit constraint $x \in \mathbb{Z}$. By stating constraint systems on real numbers, we can this way express both continuous and discrete problems, as is the case with integer or mixed models (MIP) in the extensions of linear programming (LP).

In a numeric CSP, a constraint $C(\vec{x})$ therefore identifies a subset of $\mathbb{R}^n$. Domain constraints are specific unary constraints. They bind the problem variables to their definition domain. For instance, the condition $\vec{x} \geq \vec{0}$ states a conjunction of domain constraints that restricts the problem space to the positive orthant: $x_1 \geq 0 \wedge \ldots \wedge x_n \geq 0$. The other constraints set additional conditions for a solution. Altogether, the problem restrictions define a feasible region.

By far, the best-known continuous CSP is a linear program. A linear constraint $\vec{a}\vec{x} \leq b$ (where $\vec{a}\vec{x}$ denotes a scalar product) cuts down an entire half-space. As stated in Table 1 in matrix notation, a conjunction of linear inequalities delimits a convex polyhedron.

Table 1: A linear constraint system viewed as a numeric CSP

$A\vec{x} \leq \vec{b}$	$C_1(\vec{x}) \wedge \ldots \wedge C_m(\vec{x})$
$\vec{x} \geq \vec{0}$	$\vec{x} \in D = D_1 \times \ldots \times D_n$

Solving a CSP means finding a solution or proving that the feasible region is empty. Clearly, the method of choice for solving a linear program is to use a linear solver (the satisfiability problem is of polynomial complexity). In the CSP framework however, we can easily express mixed-integer problems (including NP-complete ones) as well as non-linear problems for which solution procedures may be lacking. Therefore, deciding — in reasonable time — whether a solution exists to the problem modeled is a task that no inference engine is capable of, in general.

We can relax these requirements, either by restricting the constraint language, or by approximating the solution process. Constraint programming (CP) favors the latter option, in order to preserve the richness of CSPs. The key idea behind interval solving (which generalizes finite-domain solving) is to quickly exclude from tests areas void of solutions, so to concentrate the search effort on promising areas.

In some special circumstances, those consistency checks can be proved sufficient. For instance, to guarantee the tractability of a combinatorial search problem, it is enough to observe that the constraints are min-closed (or, symmetrically, max-closed). These algebraic properties have been introduced by Jeavons and Cooper (1995) to the study of finite-domain CSPs. But their definition is relevant to numeric CSPs as well.

In this paper, we prove that, in the general continuous case, min-closed CSPs have a 'close approximation property' from which the completeness of the specific discrete case derives. In reference to the shaving procedure used for trimming variable domains, we call it the *close shaven property*. Indeed, the 'box relaxation' an interval solver computes cannot be better refined with respect to its bottom corner, so that shaving is of no use.

Identifying min-closed constraints allows us to spot 'easy to solve' qualitative or quantitative problems. We get a certificate of tractability or, at least, of good convergence.

This research note is organised as follows. After recalling the basics of interval solving, we focus on the outcome of propagation — the underlying procedure used for pre-solving a CSP. We first characterize the reduced domains computed in the CP relaxation. Then, combining this knowledge with the definition of min-closed systems, we put forward and prove the close-shaven property that the reduced domains exhibit. We finally present a few application examples, crossing the lines between different disciplines.

2 Interval approximations in Constraint Logic Programming

When seeking solutions to numerical constraints, exact methods of computer algebra often appear too specialized or too costly. Interval approximation methods provide an alternative means which, though usually weak, is general-purpose (Older and Vellino 1993; Van Hentenryck et al. 1997; Benhamou and Granvilliers 2006).

$$-\infty < -f_M < \cdots < -f^+ < -f < \cdots < 0 < \cdots < f < f^+ < \cdots < f_M < +\infty$$

The finite numerical scale $\mathbb{F}$.

2.1 Floating point intervals

Calculations on reals are performed using a slide rule having a finite set $\mathbb{F}$ of graduations. In practice, this set is given by the IEEE scale of floating point numbers. Small integers (together with the 2 infinities) are another option. Members of $\mathbb{F} \cup \{-\infty, +\infty\}$ have a direct machine representation. But most real numbers like 1/3 lie within two graduations. As a matter of consequence, operations on reals are substituted with operations on intervals.

Following the theory of Prolog IV (Colmerauer 1994), we shall distinguish between open and closed intervals. Historically, the need for expressing strict as well as non-strict inequalities derives from the universal role given in the language to the *not equal* constraint $x \neq y$ (aka *dif*). We'll call *approximation interval* any interval of reals whose bounds (when they exist) are members of $\mathbb{F}$. We'll use the notation $[1, 2[$ equivalent to $[1, 2)$ for the half-closed and half-open interval $\{x \in \mathbb{R} \mid 1 \leq x < 2\}$. The choice of $\mathbb{F}$ leads to a finite partition[1] of the real line $\mathbb{R}$ into *atomic intervals* of the following kind:

- degenerate closed intervals $\{f\}$ reduced to a point which is a member of $\mathbb{F}$
- open intervals $]f, f^+[$ between reals that are consecutive members of $\mathbb{F}$
- the open half-line $]f_M, +\infty[$ where f_M is the greatest real member of $\mathbb{F}$
- the open half-line $]-\infty, -f_M[$ where $-f_M$ is the least real member of $\mathbb{F}$.

Any non-empty approximation interval D therefore uniquely decomposes into a disjoint union of atomic intervals that forms a totally ordered sequence. We'll call the first element Δ of that sequence the *leftmost slice* of D (and the last, its *rightmost slice*).

2.2 Interval constraints

In the sequel, we'll make the non-restrictive assumption that variable domains are convex[2]. Thus, a *domain constraint* will be a relation of the form $x \in D$ where D is an approximation interval (possibly empty). Domain constraints are handled apart in CP.

Assuming a fixed number of unknowns, say n, the conjunction of domain constraints $(x_1 \in D_1) \wedge (x_2 \in D_2) \wedge \ldots \wedge (x_n \in D_n)$ is equivalent to $\vec{x} \in D_1 \times D_2 \times \ldots \times D_n$. The cross-product of the variable's domains defines a *box*, i.e., a hyper-rectangle aligned with the axes of the Cartesian coordinate system.

A numerical constraint solver comes equipped with a catalogue of *primitive constraints* (from arithmetic, trigonometry, ...) that can be used for model building.

For the sake of simplicity, we shall assume that all the constraints used in our model are n-ary[3]. Thus, a system of m simultaneous constraints can be condensed as:

$$C_1(\vec{x}) \wedge ... \wedge C_m(\vec{x}) \wedge (\vec{x} \in D_1 \times \ldots \times D_n)$$

where each C_i denotes a primitive constraint $(1 \leq i \leq m)$ and each x_j refers to a numerical unknown $(1 \leq j \leq n)$. A constraint system is satisfiable if it has solutions. Obviously, the domain product $D_1 \times \ldots \times D_n$ provides an enclosure of all of the numerical solutions sought in $\mathbb{R}^n$. But even with a tight enclosure, a solution set may remain infinite.

[1] If the approximation is restricted to closed intervals, as in (ECLiPSe), we have a covering instead.
[2] The rationale (cf. leftmost slice) is the same for domains made of unions of approximation intervals.
[3] A relation is always the projection of its cylindrical extension.

The unary constraint 'is an integer' is of particular interest for making the connection with finite-domain constraint solving. By typing all of (or some of) the variables as integers, we are allowed to express purely combinatorial or mixed problems.

3 Interval solving

Basically, constraint solving proceeds by transforming a system of primitive constraints:

$$C_1(\vec{x}) \wedge ... \wedge C_m(\vec{x}) \wedge (\vec{x} \in D_1 \times \ldots \times D_n) \tag{1}$$

into an equivalent system:

$$C_1(\vec{x}) \wedge ... \wedge C_m(\vec{x}) \wedge (\vec{x} \in \underline{D}_1 \times \ldots \times \underline{D}_n) \tag{2}$$

where all the domains are reduced (i.e., their bounds are updated): $\underline{D}_j \subset D_j$ $(1 \leq j \leq n)$.

By reducing domains, interval *propagation* amounts to drawing valid inferences locally on variables' domains that are shared globally. Every consequence derived is correct, but not all expected consequences are derived. So, reasoning is sound but incomplete. Interval solving is akin to a pre-processing step: it often has to be complemented by search.

When domains are finite, a complete exploration of the search space is possible (at an exponential worst cost in the number of variables). When domains are infinite, search is inevitably limited by the precision of the grid being used for the calculations.

3.1 Properties of reduced domains

Since implementations of constraint solvers vary with respect to the *consistency checks* performed during propagation, we'll stick to the theory of Prolog IV to make things clear[4]. Formally, for every constraint C_i $(1 \leq i \leq m)$, the following property holds:

$$\underline{D}_1 \times \ldots \times \underline{D}_n = red_{C_i}(\underline{D}_1 \times \ldots \times \underline{D}_n) \tag{3}$$

where $red_C(D_1 \times \ldots \times D_n)$ is defined as the smallest box of $\mathbb{R}^n$ contained in $D_1 \times \ldots \times D_n$ and containing the graph of the relation C, i.e., $\{\vec{x} \in \mathbb{R}^n |\ C(\vec{x}) \wedge\ \vec{x} \in D_1 \times \ldots \times D_n\}$. The narrowing operator red_{C_i} attached to the primitive constraint C_i minimizes the size of the domains. Those are trimmed as much as possible by performing bounds' updates.

The fixed point equation (3) means interval solving is complete for a 1-constraint CSP: $C_i(\vec{x}) \wedge\ \vec{x} \in D_1 \times \ldots \times D_n$ (in case of inconsistency, the box becomes the empty set). Moreover, if the box reduces to a singleton $\{\vec{x}\}$, we have the guarantee that $C_i(\vec{x})$ holds. From this, we recover the well-known global properties of interval solving:

- There are no solutions outside the box determined by the reduced domains.
- When the enclosing box becomes empty, system (2) is equivalent to the constraint false. We thus have a proof that system (1) has no solution.
- When the enclosing box reduces to a single point, we know for sure that this point satisfies every constraint. It is therefore a solution to the whole constraint system (the unique one indeed in the original domains).

[4] Refer to Collavizza et al. 1999 or Chiu Wo Choi et al. 2006 for a comparison with stronger forms of inference, over infinite and finite domains respectively.

3.2 Bounds-consistency property

Let us examine in detail what happens when the system (2) is *locally consistent*, i.e., when none of the reduced domain is empty. Then, every $\underline{D}_j$ has a leftmost slice, Δ_j. According to (3), for every constraint i (taken independently), and for every variable j, the following formula holds — otherwise, the slice Δ_j would have been ruled out:

$$\exists \vec{x}\ C_i(\vec{x}) \wedge (\vec{x} \in \underline{D}_1 \times \cdots \times \Delta_j \times ... \times \underline{D}_n) \tag{4}$$

When all of the reduced domains are bounded and closed — as is the case with finite domains, we retrieve the classical definition of *bounds-consistency*, also known as 2B- (Lhomme 1993) or hull-consistency (Benhamou et al. 1999). We can always find in the box $\underline{D}_1 \times \cdots \times \underline{D}_n$ a solution to $C_i(\vec{x})$ extending the partial assignment $x_j = min(\underline{D}_j)$, since for every constraint i ($1 \leq i \leq m$) and every domain j ($1 \leq j \leq n$) we have:

$$\exists \vec{x}\ C_i(\vec{x}) \wedge (\vec{x} \in \underline{D}_1 \times \cdots \times \{min(\underline{D}_j\} \times ... \times \underline{D}_n) \tag{5}$$

A similar analysis applies to rightmost slices and maximum domain values.

3.3 Variants of primitive constraints *(can be omitted in a first reading)*

Quite often, quantitative relationships are specified using mathematical functions. We can define for instance two constraints over $\mathbb{R}^n$ out of a function f, from $\mathbb{R}^{n-1}$ into $\mathbb{R}$: $C(\vec{x}, y)$ if and only if $y = f(\vec{x})$ and $C'(\vec{x}, y)$ if and only if $y \geq f(\vec{x})$. The definitions of C and C' being fairly close, the following lemma shows we don't need to implement two narrowing operators, red_C and $red_{C'}$. We can define C' as a variant of the primitive constraint C, without loosing the native properties (3, 4) that propagation guarantees.

Lemma *Let C be a primitive constraint over $\mathbb{R}^n$ and Op an inequality symbol ($<$, $\leq$, $\geq$, or $>$). Define C' in terms of C by $C'(\vec{x}, y) = \{(\vec{x}, y) \in \mathbb{R}^n \mid \exists z\ C(\vec{x}, z) \wedge (z\ Op\ y)\}$. Then, the bounds-consistency property holds for the defined constraint C'.*

Proof
Consider a system made of the pair of constraints defining C', together with some domain restrictions (without loss of generality, we here assume the inequality stated is $z \leq y$):

$$C(\vec{x}, z) \wedge (z \leq y) \wedge (\vec{x} \in D_1 \times \ldots \times D_{n-1}) \wedge (y \in D_n) \wedge (z \in D_{n+1})$$

Let $\underline{D}_1 \times \ldots \times \underline{D}_n \times \underline{D}_{n+1}$ be the cross-product of the reduced domains. Assume the system is locally consistent and suppose $\underline{D}_1 \times \ldots \times \underline{D}_n$ is not the smallest box containing the solutions to the system $C'(\vec{x}, y) \wedge (\vec{x} \in D_1 \times \ldots \times D_{n-1}) \wedge (y \in D_n)$. Then, we can safely cut away a leftmost or rightmost slice, Δ_i, from a reduced domain $\underline{D}_i$ ($1 \leq i \leq n$). The constraint $C'(\vec{x}, y)$ having no solution in the box $\underline{D}_1 \times \ldots \times \Delta_i \ldots \times \underline{D}_n$, there exists no value of z for which the system $C(\vec{x}, z) \wedge (z \leq y)$ is satisfiable, and all the more so if z lies in the rightmost slice Δ_{n+1} of $\underline{D}_{n+1}$. Still, the bounds-consistency property holds for C in the conjunct, so $C(\vec{x}, z)$ remains satisfiable with $(\vec{x}, y, z)$ in $\underline{D}_1 \times \ldots \times \Delta_i \ldots \times \underline{D}_n \times \Delta_{n+1}$. We thus have a particular solution to C, say $(\vec{x}^*, z^*)$. The same is true for the inequality $z \leq y$. Observe that the leftmost slice of the reduced domain of y can only be equal to Δ_{n+1} or follow it, so that we can always choose a value $y^* \geq z^*$. We are thus able to construct a point of $\underline{D}_1 \times \ldots \times \Delta_i \ldots \times \underline{D}_n \times \Delta_{n+1}$ that is at the same time a solution to $z \leq y$ and a solution to $C(\vec{x}, z)$. Hence a contradiction. $\square$

4 Min-closed constraint systems

4.1 Definitions

We say that a subset S of $\mathbb{R}^n$ is closed for the minimum operation (in short, *min-closed*) if, whenever $\vec{x}$ and $\vec{y}$ are in S, their minimum (defined component-wise) also lies in S:

$$\vec{x} \in S \wedge \vec{y} \in S \implies min(\vec{x}, \vec{y}) = (min(x_1, y_1), \ldots, min(x_n, y_n)) \in S \tag{6}$$

By definition, a constraint is min-closed if its solution set is min-closed.

A conjunction of min-closed constraints forms a min-closed system. Its solution set is min-closed, for the reason that the intersection of min-closed sets is min-closed. Similarly, the projection onto a subset of variables of a min-closed system is min-closed.

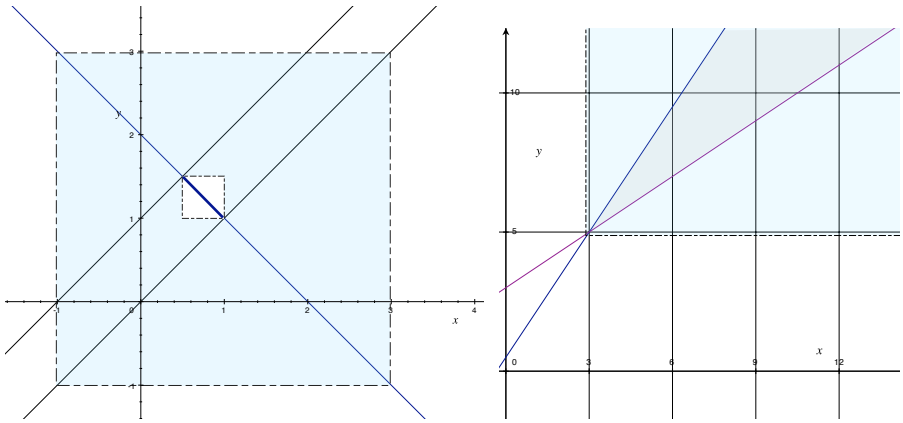
4.2 First catalogue of min-closed constraints

- Unary constraints are min closed. In particular, domain constraints are min-closed.
- Binary constraints of the form $y = f(x)$ are min-closed if f is a monotonically increasing function of x, since $f(min(x_1, x_2)) = min(f(x_1), f(x_2))$.

Table 2: Min-closed primitive (or near-primitive) constraints.

	Constraint		**Reified constraint**	
Comparison	$x = c$	•		
	$x \neq c$	•		
	$x \leq c$	•		
	$x < c$	•		
	$x \geq c$	•	$\delta \equiv (x \geq c)$	•
	$x > c$	•	$\delta \equiv (x > c)$	•
	$x = y$	•		
	$x \leq y$	•		
	$x < y$	•		
Integrity	$x \in \mathbb{Z}$	•		
Boolean	$\neg\beta$	•		
	$\beta \wedge \gamma$	•	$\delta \equiv (\beta \wedge \gamma)$	∘
	$\beta \Rightarrow \gamma$	•		
Linear	$x + c = y$	•		
	$x - c = y$	•		
	$x + y \leq z$	∘		
	$x + y < z$	∘		
	$ax = y$	•		
Non-linear	$xy \leq z,\ x \geq 0,\ y \geq 0$	∘		
	$y = sqrt(x)$	$\{(x, y) \in \mathbb{R}^2 \mid y = \sqrt{x},\ x \geq 0\}$	•	
	$y = log(x)$	$\{(x, y) \in \mathbb{R}^2 \mid y = \log x,\ x > 0\}$	•	
	...			

All of the constraints in this excerpt are min-closed. Some are max-closed too (• mark).
Notations: x, y, z are real variables; β, γ, δ binary variables, i.e., restricted to {0,1}; c is a numerical constant; a a non-negative coefficient.



(a) A counter-example (line segment) (b) A min-closed polyhedron

Fig. 1: Linear constraint systems and their CP relaxations (boxes)

5 The 'close-shaven' property

Interval solvers are notoriously bad at solving linear systems. Contrary to linear solvers, they miss a global view. Fig. 1a gives an example where propagation does not perform any pruning at all. The CSP is: $(-1 \leq x \leq 3) \wedge (-1 \leq y \leq 3) \wedge (x \leq y) \wedge (y \leq x+1) \wedge (x+y = 2)$. The *CP relaxation* $[-1, 3] \times [-1, 3]$ leads to a quite pessimistic approximation of the feasible region (highlighted line segment) whose exact projections are [0.5, 1] and [1, 1.5].

When the overall interval approximation is too loose, *shaving* heuristics can be used for tightening bounds without creating choice points (contrary to general search procedures). Shaving is therefore deterministic and attempts to exclude a leftmost (or rightmost) part of a domain as long as it is safe do so (i.e., when its cross-product with the other domains is void of solution). In the worst case, shaving may have to recursively examine all the atomic boxes that decompose a domain. So, shaving can be potentially costly too.

5.1 General continuous case

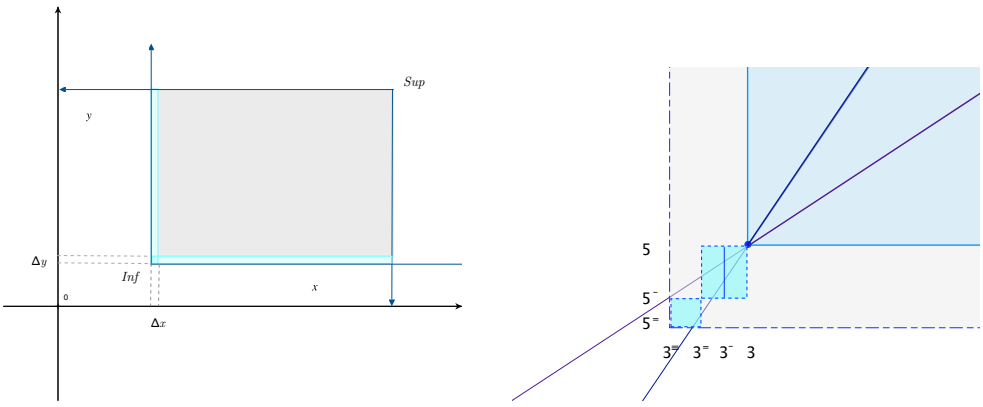
We are now at a point where we can state the approximation property that min-closed systems show for interval solving.

Proposition *Assume $C_1(\vec{x}) \wedge ... \wedge C_m(\vec{x}) \wedge (\vec{x} \in \underline{D}_1 \times \ldots \times \underline{D}_n)$ is a min-closed system that is locally consistent. Then, considering the leftmost slices Δ_j of the reduced domains $\underline{D}_j$, the system $C_1(\vec{x}) \wedge ... \wedge C_m(\vec{x}) \wedge (\vec{x} \in \Delta_1 \times \ldots \times \Delta_n)$ is also locally consistent.*

Proof
According to the bounds-consistency property (4), for each constraint C_i $(1 \leq i \leq m)$, we have a solution in $\Delta_1 \times \underline{D}_2 \times \ldots \times \underline{D}_n$, a solution in $\underline{D}_1 \times \Delta_2 \times \ldots \times \underline{D}_n$, ... and a solution in $\underline{D}_1 \times \ldots \times \underline{D}_{n-1} \times \Delta_n$. Their minimum is a point in $\Delta_1 \times \ldots \times \Delta_n$. Since the constraint is min-closed, it is a solution to C_i.

So, each constraint C_i is satisfiable within the cross-product $\Delta_1 \times \ldots \times \Delta_n$ which defines an atomic box and cannot be further reduced. It follows that the constraint system $C_1(\vec{x}) \wedge ... \wedge C_m(\vec{x}) \wedge (\vec{x} \in \Delta_1 \times \ldots \times \Delta_n)$ is locally consistent. $\square$

Consequently, none of the lower bounds of the reduced domains obtained by propagation can be further improved by shaving: the reduced domains are all *close-shaven*. Still, the system may be globally inconsistent, but bisection then is of no help. The solver cannot disprove the formula: $\exists \vec{x}\ C_1(\vec{x}) \wedge ... \wedge C_m(\vec{x}) \wedge (\vec{x} \in \Delta_1 \times \cdots \times \Delta_j \times \cdots \times \Delta_n)$.



(a) The leftmost slices Δ_x and Δ_y in a 2D case. (b) Zoom in on $\Delta_x \times \Delta_y$ for Fig. 1b example.

Fig. 2: Search space decomposition into atomic slices and boxes

When constraints are min-closed, one should not observe gross over-approximations in the CP relaxation. By way of illustration, the cone of Fig. 1b is defined by two linear inequalities in the $\mathbb{R}^2$ plane: $(3x \geq 2y-1) \wedge (3y \geq 2x+9)$. This CSP is min-closed. It has a least element: $(x, y) = (3, 5)$. The close-up on Fig. 2b shows that the minimum, though not exactly computed, is precisely surrounded by the infimum of the reduced domains. $3^{\equiv}$ denotes the third float preceding 3 and $5^{=}$ the second before 5. Atomic boxes that are filled indicate areas where the interval solver cannot exclude solutions. We see in particular that the leftmost slices of the two domains cannot be shaved off.

5.2 Specific discrete case

With finite domains, we derive from the close-shaven property a least element solution (minimum of the box). This witness point ensures satisfiability and restores decidability.

Corollary *Consider a min-closed system whose domains are bounded from below and whose variables are of integer type . If the reduced domains are not empty, then the CP relaxation $\underline{D}_1 \times \ldots \times \underline{D}_n$ has a least element which is a solution to the constraint system.*

Proof
Since the domains are closed and non empty, they have a minimum which is also their lowest slice. The atomic box $\Delta_1 \times \cdots \times \Delta_n$ is a singleton point of integer coordinates $min(D_1), \ldots, min(D_n)$. As it satisfies every constraint, it is a solution. □

We retrieve the fact that propagation (which is a polynomial-time procedure) is complete for min-closed finite-domain satisfiability checking[5]. Such CSPs are thus tractable.

[5] The original paper of (1995) proved that a min-closed CSP is solved by enforcing arc-consistency only, a stronger form of inference (Mackworth) than bounds-consistency. But obviously, the latter suffices.

6 Cross-connections and examples

6.1 Application to linear systems

The following result generalizes the case illustrated by Fig. 1b.

Property 1 *Every linear constraint of the form $a_1x_1 + \ldots + a_nx_n \leq cy + b$ where the a_i's are non-negative is min-closed.*

This 'meta-constraint' easily decomposes into a min-closed system, using the primitives of Table 2. If we define a *quasi-positive* matrix as a matrix having at most one negative entry per row, it follows that a linear system $A\vec{x} \leq \vec{b}$ is min-closed if A is quasi-positive. Then, if the feasible region is non-empty and bounded from below, it has a least element.

Now, there is a result from Cottle and Veinott (1972) stating exactly the converse:

Property 2 *If A is quasi-positive, a convex polyhedron of the form $P = \{\vec{x} \geq 0 \mid A\vec{x} \leq \vec{b}\}$ has a least element if it is non-empty.*

So, for linear systems we see here a one-to-one correspondence between the algebraic property (min-closed system), the geometric property (existence of a minimum) and the syntactic property (quasi-positive matrix).

Furthermore, we have a special case ensuring integrality (Chandrasekaran 1984):

Property 3 *Assuming that A and $\vec{b}$ have integer coefficients, if A is quasi-positive and if the negative entries of A are equal to -1, then the polyhedron $P = \{\vec{x} \geq 0 \mid A\vec{x} \leq \vec{b}\}$ has an integral least element if it is non-empty.*

Note this result holds for integer as well as real valued variables. It follows that propagation is complete for the numeric CSPs of that class since it involves no division. The polyhedron's least element is a corner-point for both the CP and LP relaxations.

A tension problem in a network (which is the dual of a flow problem) (Chvátal 1983) is characterized by difference bounded constraints of the form $a \leq x - y \leq b$. With 2 non-zero entries per row, equal to $+1$ and -1, its matrix is quasi-positive and quasi-negative. Such a linear system globally acts as a primitive constraint for interval solving — a feature which has direct applications to discrete- as well as continuous-time scheduling.

6.2 Application to rule-based systems

Property 4 *A conditional constraint of the form: $(x_2 \geq a_2) \wedge \ldots \wedge (x_k \geq a_k) \Rightarrow (x_1 \geq a_1)$ is min-closed.*

Again, the property holds for integer and real variables (regarding the expressivity of a knowledge engineering language, this point is worth noting). More generally, the right-hand side of the implication can be any min-closed constraint.

We proved elsewhere (2010) that a finite-domain CSP made of rules of the above form can be translated into a Horn propositional clausal system, i.e., into a well-known linearly satisfiable SAT problem. We coined the term *Horn-reencodable* for characterizing a rule base of such kind.

Now, according to Jeavons and Cooper (1995), every min-closed finite-domain constraint is logically equivalent to a conjunction of conditionals of that form. This means that a min-closed finite-domain CSP is indeed Horn-reencodable.

The ability to provide a solution by joining the minima of the reduced domains thence derives from the integer least element property of a Horn polytope (Chandru and Hooker).

6.3 Application to simulation systems

Convergence and numerical stability of the computation methods used are major concerns in Numerical analysis when solving large (often sparse) equation systems. Many come from the discretization of differential equations like the Laplacian equation for simulating heat transfer. A typical 2D pattern is : $-4x_{i,j} + x_{i,j+1} + x_{i,j-1} + x_{i-1,j} + x_{i+1,j} = 0$. It relates each cell to its north, east, south and west neighbours. We have one equation per cell, hence a system $A\vec{x} = \vec{b}$ with as many equations as unknowns (cell values).

We observe that the square matrix A is quasi-positive. It is also diagonally dominant, so that the solution we expect to be non-negative is also known to be unique when it exists (cf. cell assignments stating boundary conditions). The equation system is the conjunction of 2 systems of inequalities: $A\vec{x} \leq \vec{b}, \vec{x} \geq \vec{0}$ and $A\vec{x} \geq \vec{b}, \vec{x} \geq \vec{0}$. The former is min-closed. The latter is max-closed. Therefore, the value of the physical solution will lie between a least and a greatest element, both computed with a high quality of approximation. Moreover, if only one bound is needed, solving half of the system will do.

7 Conclusion

Compared to polyhedral studies for MIPs, there are not many theoretical results available for figuring out in advance the outcome of a CP computation. However, it is a well-known fact that two formulations that are logically equivalent are rarely equal in efficiency. For understanding a CSP behaviour, a human modeler has to master the idiosyncrasies of the type of engine he intends to use.

In this paper, we have shown with interval solving how to generalize the completeness property of propagation on min-closed systems, from finite domains to infinite domains. We have answered the question of what is the meaning of a good approximation for an interval solver. It is a box enclosure of the feasible region that cannot be refined by search. With min-closed systems, the CP relaxation delineates the best outward approximation of the infimum of the solution set. And this is obtained free of search. The same is true for max-closed systems (with respect to the supremum).

The polynomial class of min-closed problems has a theoretical significance since, among its instances, we find Horn satisfiability and Critical Path optimization problems. It also has practical applications, for instance, to machine scheduling (Purvis and Jeavons 1999) or product configuration problems (Narboni 2010).

From the point of view of complexity theory, Bodirsky and Kára (2009) have thoroughly investigated the generalization of min-closed CSPs to infinite domains for temporal constraint languages. Though restricted to logical combinations of variable comparisons, those languages can be of use for specifying continuous-time disjunctive scheduling problems, or modeling rule-based systems too. The problem being amenable to quantifier elimination, it is decidable. The authors prove a dichotomy result and provide an exact solution algorithm for the polynomial case.

Our analysis gives an intuitive insight into the building and interpretation of interval-based models. In the light of it, we may identify 'well-solved' sub-problems, occasionally predict the robustness of a solution design or assess the effectiveness of a search heuristics, prior to passing test experiments.

References

Apt, K. and Wallace, M. 2007. *Constraint Logic Programming using ECLiPSe*, Cambridge.

Benhamou, F. et al. 1995. *Le manuel de Prolog IV*, PrologIA. http:// www.prolog-heritage.org.

Behamou, F. and Goualard, F. and Granvilliers, L. and Puget, J-F. 1999. Revising hull and box consistency. In *Procs. ICLP'99*, 230-244.

Benhamou, F. and Granvilliers, L. 2006. Continuous and Interval Constraints. In *Handbook of Constraint Programming*, 571-603, Elsevier.

Bodirsky, M. and Kára, J. 2009. The complexity of temporal constraint satisfaction problems. *J. ACM*, 57:2 1-41.

Chandrasekaran, R. 1984. Integer programming problems for which a simple rounding type of algorithm works. In *Progress in Combinatorial Optimization* (Pulleyblanck Ed.), 101-106.

Chandru, V. and Hooker, J. 1999. *Optimization methods for logical inference*, Wiley.

Chiu Wo Choi and Harvey, W. and Lee, J. and Stuckey, P. 2006. Finite Domain Bounds Consistency Revisited. In *Procs. Australian Joint Conference on Artificial Intelligence*, 49-58.

Chvátal, V. 1983. *Linear Programming*, Freeman.

Cohen, D. and Jeavons, P. 2003. Tractable Constraints Languages. In *Constraint Processing* by R. Dechter, 299-331, Morgan Kaufmann.

Collavizza, H. and Delobel, F. and Rueher, M. 1999. Comparing partial consistencies. *Reliable computing*, 5:3 213-228.

Colmerauer, A. 1994. *Spécifications de Prolog IV*. Rapport technique, Faculté des Sciences de Luminy.

Cottle, R. and Veinott, A. 1972. Polyhedral sets having a least element. *Mathematical Programming*, 3 238-249.

Jeavons, P. and Cooper, M. 1995. Tractable Constraints on Ordered Domains. *AI Journal*, 79:2 327-339.

Lhomme, O. 1993. Consistency techniques for Numeric CSPs. In *Procs. of the 13th IJCAI*, 232-238.

Mackworth, A. 1977. Consistency in networks of relations. *Artificial Intelligence*, 8:1 99-118.

Narboni, G. 2010. A domain language for expressing engineering rules and a remarkable sub-language. In *Procs. IKBET ECAI'10* (A. Felferning, F. Wotowa Eds.). Journal version: 2013. On rule systems whose consistency can be locally maintained. *AI Communications*, 26:1 67-77, IOS Press.

Older, W. and Vellino, A. 1993. Constraint Arithmetic on Real Intervals. In *Constraint Logic Programming: Selected Research*, 175-195, MIT Press.

Purvis, L. and Jeavons, P. 1999. Constraint tractability theory and its application to the product development process for a constraint-based scheduler. *In Procs. of PACLP'99*, 63-79, The Practical Application Company.

Van Hentenryck, P. et al. 1997. *Numerica*, MIT Press.