Extending consistent domains of numeric CSP

Helene Collavizza, Francois Delobel, Michel Rueher
Universite de Nice-Sophia-Antipolis, I3S — ESSI
930, route des Colles - B.P. 145, 06903 Sophia-Antipolis, France
{Helen, delobel, rueher} @essi.fr

Abstract

This paper introduces a new framework for extending consistent domains of numeric CSP. The aim is to offer the greatest possible freedom of choice for one variable to the designer of a CAD application. Thus, we provide here an efficient and incremental algorithm which computes the maximal extension of the domain of one variable. The key point of this framework is the definition, for each inequality, of an *univariate* extrema function which computes the left most and right most solutions of a selected variable (in a space delimited by the domains of the other variables). We show how these univariate extrema functions can be implemented *efficiently.* The capabilities of this approach are illustrated on a ballistic example.

1 Introduction

This paper introduces a new framework for extending the domain of one variable in a consistent CSP ' which is defined by a set of *non-linear constraints over the reals.* The aim is to offer the greatest freedom of choice of possible values for a variable to the *designer* of a CAD application. For example, one starts from the knowledge of a solution and tries to widen the variations of a variable. This problem occurs in a large class of electro-mechanical engineering and civil engineering applications, where extending the domain of a variable permits the tolerance of any associated component to be enlarged, and therefore to lower the cost of this component. These problems are often under-constrained. So, what the user wants to know is a subset of the solutions. For these applications, classical methods (e.g.,[7; 10]), based on local consistencies and domain splitting, cannot ensure that a solution exists inside the arbitrarily small intervals they compute. Moreover, domain splitting is ineffective if the solution set is not a finite set of isolated solutions but a collection of intervals.

The framework we introduce here allows one to enlarge the domain of a variable while preserving the consistency of the CSP. Sam-Haroud and Faltings [9] have proposed an approach for computing safe solutions of non-linear constraint systems. Roughly speaking, they fill up the solution space with a set of consistent boxes[2]. Their approach could be used to extend the domain of one variable. However, the underlying costs in computation time and space are exponential.
The framework we introduce here is less general but it can be implemented efficiently. Before going into the details, let us outline our framework in very general terms. The main steps of the right extension[3] of the domain of a variable are:

1. Searching for a subset of the solution space; this solution space may be reduced to a single point;
2. Selecting of the variable the domain of which has to be extended;
3. Defining for each inequality of an extrema function that computes the left most solution of the selected variable in a space delimited by the domains of the other variables;
4. Finding the smallest solution of all extrema functions.

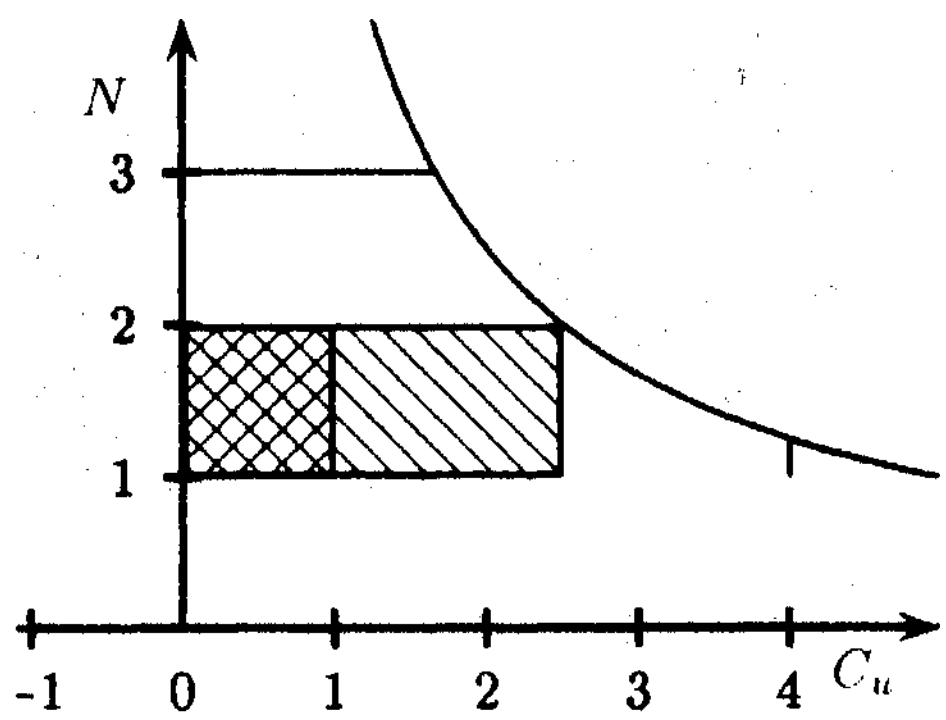
The following example illustrates this process.

[1] *An* introduction to CSP and numeric CSP can be found in [4; 7].

"Their approach is based upon a classical method used in graphical computing for image synthesis (composition of shapes, of scenes) known as the 2^k trees. The key idea is to classify portions of space in three categories: the black shapes contain no solution at all, the gray shapes contain solutions, but also contain points which .are not solutions, and finally, the white shapes contain only points which are solution. The gray shapes are split into smaller one that are again classified into black, white and gray shapes; the decomposition process stops when the size of the shapes becomes smaller than a given value.

[3]Throughout this paper, we will only consider the right extension since the left one can be computed in a symmetrical way.

Example 1 *Let us consider the behavior of an electrical shunt motor, the speed of which may be changed. The maximum speed can be up to 3 times the value of the minimum speed. We only consider two parameters of the motor: the torque C_u, and the rotation speed N. The motor cannot use more than a given power :* $N * C_u \leq P_{max}$. *Moreover, the motor cannot operate above a given speed and torque:* $N \in [1,3]$, $C_u \in [0,4]$
We know that the motor is working efficiently for every tuple of values $D_{C_u} \times D_N$ *in* $[0,1] \times [1,2]$ *when* $P_{max} = 5$.
What we want to compute is the maximum range of values of the torque which is safe with this motor. In other words, we are looking for the maximum domain D_{C_u} *such that every tuple* $D_{C_u} \times D_N$ *is a solution of the constraint system.*
Now, consider equation $N * C_u = 5$ *in the space delimited by* $N \in [1,3]$, $C_u \in [0,4]$. *Its left, most solution is the point defined by* $C_u = 2.5$ *and N = 2; this point is obviously an upper bound of the. domain* D_{C_u}



Figure[1] 1: Relation between TV and C_u

An initial subset of the solution space can often be found by experimentation. Note that the solution space may be reduced to a single point and the domains of the different variables mav successively be extended.
The definition of the univariate extrema functions is a key point of our approach. Optimal univariate extrema functions can trivially be computed for the so-called primitive constraints. For non-primitive constraints, the methods used for computing Box-consistency [l] provide an efficient way to compute a safe approximation of univariate extrema functions.
To define formally the extension of the domain of a variable, we introduce an "internal" consistency, named *in-consistency,* which ensures that every tuple in the Cartesian product of the variable domains is a solution of the constraint system, i-consistency should not be mistaken with arc consistency or approximations of arc consistency [3] (e.g. 2B-consistency [7], Box consistency[l]). Those consistencies define regions containing all the solutions (and possibly tuples which are not solution) whereas i-consistency defines a region which is a subset of the set of solutions. Figure 2 shows the relations between these different families of consistencies. Roughly speaking, the smallest external box is the best approximation which can be computed by approximations of arc consistency over continuous domains.

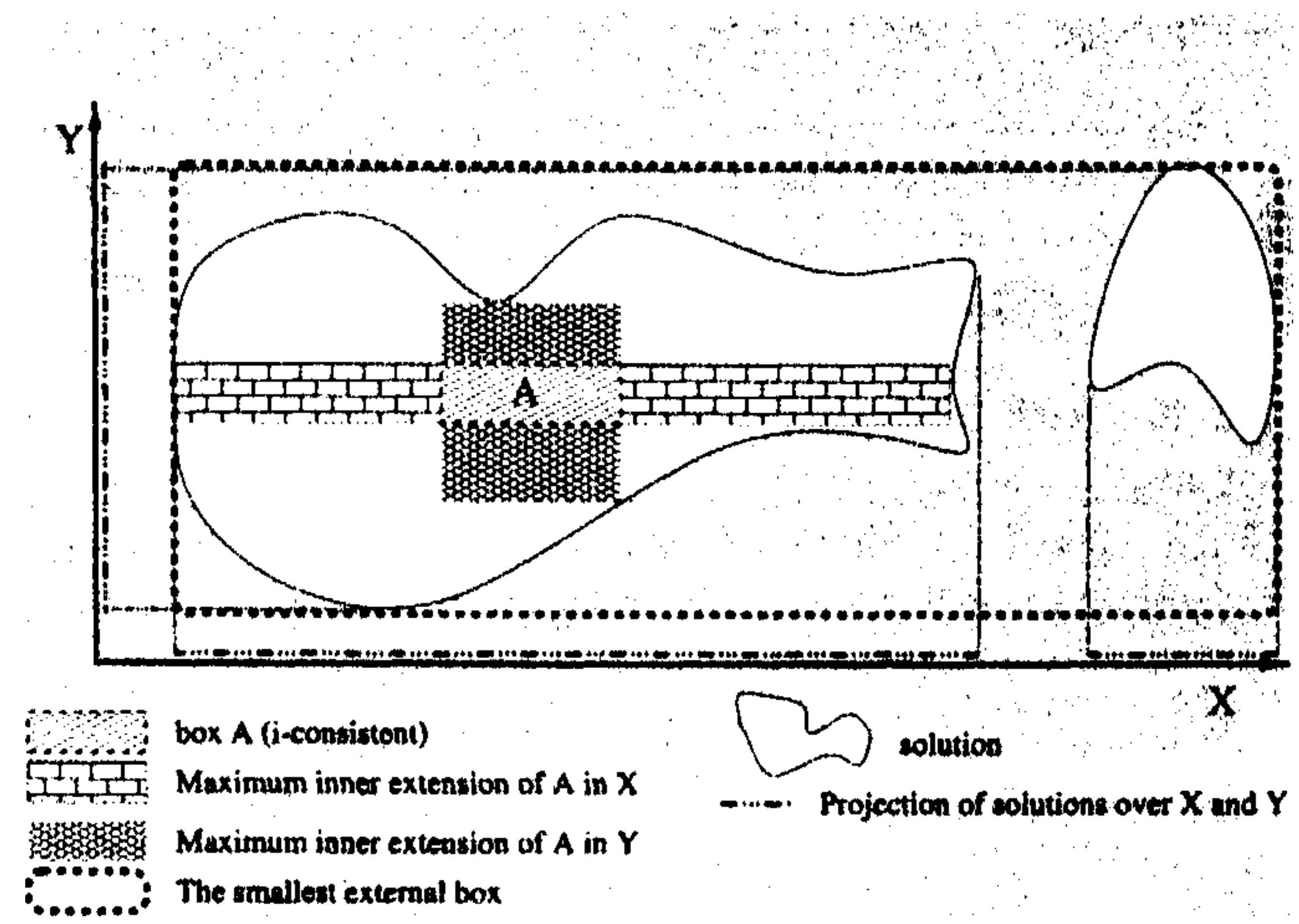


Figure 2: Relations between i-consistency and some partial consistencies

Outline of the paper: Section 2 introduces the notation and recalls the basics on CSP over continuous domains which are needed in the rest- of the paper. Section 3 is devoted to the description of the i-consistent extension process. Extrema functions are formally defined and an efficient algorithm is introduced. Section 4 outlines the capabilities of our approach on a ballistic example.

2 Preliminaries

2.1 Notation

We use the following notations, possibly subscripted:

- *x,y,z* denote variables over the reals;
- *u, v* denote real constants;
- *f,g* denote functions over the reals;
- *c* denotes a constraint over the reals;

The next subsection recalls a few notions of numeric CSP; Details can be found in [2; 10; 3].

2.2 Interval constraint system

A K:-ary constraint *c* is a relation over the reals.

Definition 1 (Interval) *Let* $\mathbb{F}$ *denote a finite subset of* $\mathcal{R}$ *augmented with the two infinity symbols* $\{-\infty, +\infty\}$. *An interval* $[a,b]$ *with* $a, b \in \mathbb{F}$ *is the set of real numbers* $\{r \in \mathcal{R} \mid a \leq r \leq b\}$.

Definition 2 (CSP)
A CSP [8] is a triple $(\mathcal{X},\mathcal{D},\mathcal{C})$ *where* $\mathcal{X} = \{x_1, \ldots, x_n\}$ *denotes a set of variables,* $\mathcal{D} = \{D_{x_1}, \ldots, D_{x_n}\}$ *denotes a set of domains,* D_{x_i} *being the interval containing all*

acceptable values for x_i, and $\mathcal{C} = \{c_1, \ldots, c_m\}$ denotes a set of constraints[4].

$\mathcal{D}$ denotes the Cartesian product $D_{x_1} \times \ldots, \times D_{x_n}$. $\vec{v}$ denotes the tuple $(v_{x_1}, \ldots, v_{x_n})$ such that $\vec{v} \in \vec{\mathcal{D}}$. $\pi_x(\vec{v})$ denotes the projection over x of $\vec{v}$. $\underline{D_i}$ (resp. $\overline{D_i}$) denotes the lower bound (resp. upper bound) of the interval D_i.

Definition 3 (k-box) *A k-box $I_1 \times \ldots \times I_k$ is the part of a k-dimension space defined by the Cartesian product of intervals $(I_1, \ldots, I_k)$*

By construction, all the k-boxes are convex.

2.3 Local consistencies

Local consistencies over continuous domains are based on arc consistency[8] which was originally defined for finite domains. This section introduces two local consistencies that will be used in the rest of the paper.

Definition 4 (arc consistency) *A CSP P $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ is arc-consistent iff: $\forall D_x \in \mathcal{D}$, $\forall v_x \in D_x$, $\forall c \in \mathcal{C}$: $\exists\, \vec{v} \in \vec{\mathcal{D}} \mid \pi_x(\vec{v}) = v_x \wedge c(\vec{v})$*

Davis ([4]) has studied the application of the Waltz algorithm ([12]) over continuous domains and has shown important theoretical limitations. The Waltz algorithm was then extended by Faltings ([5; 6]) in order to deal with ternary constraints defined by continuous and differentiable curves.

Definition 5 (Set Extension) *Let S be a subset of R. The approximation of S —denoted* hull— *is the smallest interval I such that $S \subseteq I$*

2.4 Box-consistency

Roughly speaking, Box-consistency [l; 10] is a local consistency over continuous domains which computes a safe approximation of the solution of each variable involved in a given constraint.

Definition 6 (Box–consistency) *Let $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ be a CSP and $c \in \mathcal{C}$ a k-ary constraint over the variables $(x_1, \ldots, x_k)$. c is Box-consistent if, for all x_i in $\{x_1, \ldots, x_k\}$ such that $D_{x_i} = [a, b]$, the following relations hold :*
1. $C(D_{x_1}, \ldots, D_{x_{i-1}}, [a, a^+), D_{x_{i+1}}, \ldots, D_{x_k})$,
2. $C(D_{x_1}, \ldots, D_{x_{i-1}}, (b^-, b], D_{x_{i+1}}, \ldots, D_{x_k})$.
where a^+ (resp. a^-) corresponds to the smallest (resp. largest) number of $\overline{\mathbb{F}}$ strictly greater (resp. smaller) than a, and C stands for interval extension of c [3].

The essential point is that the variable x is Box-consistent for constraint $f(x, x_1, \ldots, x_n) = 0$ if the bounds of the domain of x correspond to the leftmost and the rightmost 0 of the optimal interval extension of $f(x, x_1, \ldots, x_n)$.

[4]It is worthwhile to notice that the set of constraints C represents a conjunction of constraints that have to be satisfied. Disjunctions may only occur inside a single constraint, e.g. the single constraint $x^2 = y$ is equivalent to the disjunction $(x = \sqrt{y}) \vee (-x = \sqrt{y})$.

3 Extension of the domain of a variable of a CSP

This section introduces the way a domain of a single variable can be extended while preserving consistency of the whole CSP. We start by defining two local consistencies which are needed to characterize the extended domains.

Next, we formally define the univariate extrema functions that actually compute the bounds of the i-consistent extensions of the domain of a variable.

3.1 e-consistency

Various approximations of arc consistency (e.g. 2B-consistency[7], Box consistency[l]) have been introduced for continuous domains, e consistency is the best approximation of the solution space which can be computed by these partial consistencies. For instance, e consistency corresponds to the "smallest external box" on Fig. 2. More formally, e-consistency is defined as follows:

Definition 7 (e–consistency) *Let $P = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ be an arc consistent CSP. A CSP $P' = (\mathcal{X}, \mathcal{D}', \mathcal{C})$ is e consistent iff $\forall D'_i \in \mathcal{D}' : D'_i =$* $\texttt{hull}(D_i)$

In other words, a CSP P = (X,D,C) is e-consistent iff P' = $(\mathcal{X}, \mathcal{D}', \mathcal{C})$ is arc consistent and V corresponds to the smallest box containing all values of D'. So, for inequality c, e-consistency on the corresponding equation c_{equ} (see section 3.4) yields a box which bounds the maximal extension that can be performed for any variable occurring in *c*.

3.2 i~consistency

Definition 8 (i -consistency)
Let $P = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ be a CSP. P is i-consistent iff $\forall c \in \mathcal{C}, \forall \vec{v} \in \vec{\mathcal{D}}$: $c(\vec{v})$

In other words, a CSP P — (X,D,C) is i consistent iff V only contains tuples which are solutions.

Example 2 *Let P be the CSP defined by $\mathcal{X} = \{x, y\}$, $D_x = [-2, 2], D_y = [6, 12]$, $\mathcal{C} = \{x^2 - y \leq 0, -x^2 + 20 \leq y\}$. This system is i consistent :*
$\forall (v_x, v_y) \in (D_x, D_y)$: $v_x^2 - v_y \leq 0 \wedge -v_x^2 + 20 \leq v_y$
Now, we want to find the largest $D'_x \supseteq D_x$ such that the CSP defined by replacing D_x by D'_x in P is i-consistent.

Figure 3 shows the original box and the extension of D_x. Both boxes are i-consistent[5]. *The domain of y remains unchanged. B is the e–consistent box for equation $x^2 = y$ and gives the upper bound of the extension of D_x.*

Ward et al. [13] have proposed four kinds of interval propagation. One of them is related to i-consistency. Each interval D_x is labeled with one of these kinds:

[5]Note that we could also perform a fruitful i- consistent extension of D_y to [6,14] with the new box. But this extension of D_y is much smaller than the one we would have obtained if we had extended the initial box (D_y would have been extended to [4,16]). In general, the result of successive extensions by i-consistency of several variables depends on the processing order of the variables.

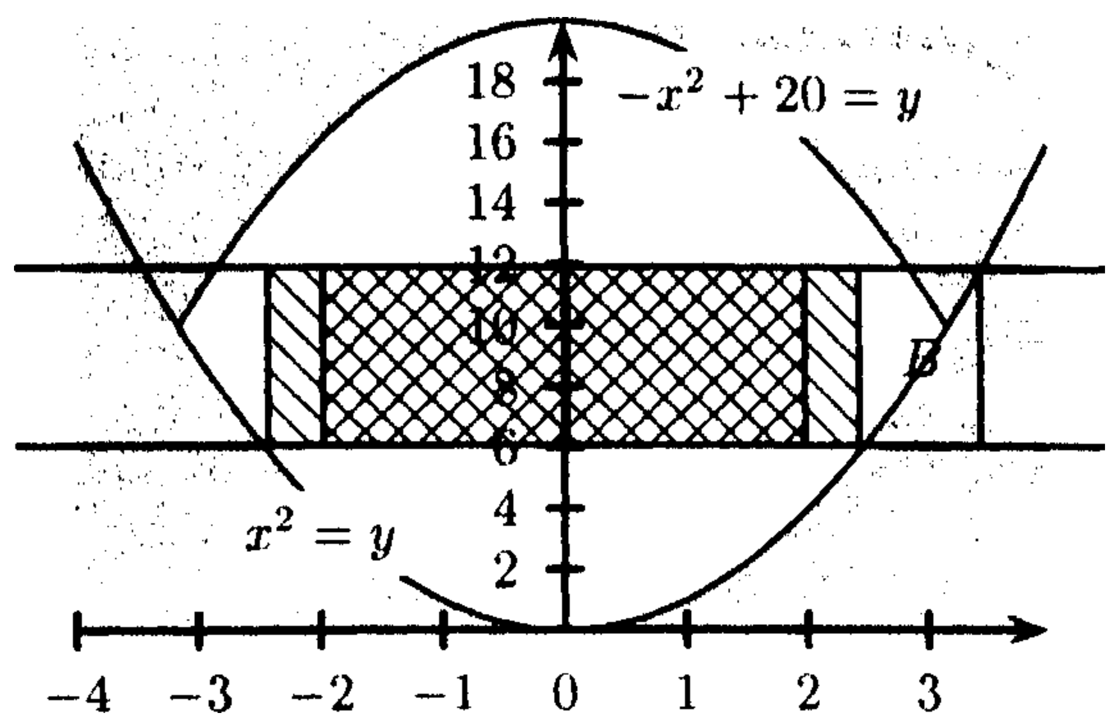


Figure 3: Maximal i-consistent extension of D_x

"only", "every", "some" and "none". If D_x is labeled "only" then solution tuples only take their values for x in D_x . If D_x is "every" then every value of x in D_x gives a tuple solution. If D_x is "some" then there exists at least one solution tuple such that x takes its value in D_x. If D_x is "none" then there is no solution tuple such that the value of x is in D_x

Labelling every variable with "every" is what we call i-consistency. However, Ward et al.'s inference rules that allow computing labelled interval propagation do not consider the case where two variables are labelled "every". Moreover, these inference rules assume strong monotony and continuity properties of the constraint system.

Now, we formally define what we mean by right i-consistent extension.

3.3 Right i-consistent Extension of D_x

Definition 9 (Right i consistent extension of D_x)

Let $P = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ be a i-consistent CSP. $P' = (\mathcal{X}, \mathcal{D}', \mathcal{C})$ is a right i-consistent extension of D_x for P iff:

- $\forall D_i \in \mathcal{D} \setminus \{D_x\},\ D_i = D'_i$
- $D_x \subset D'_x,\ \underline{D'_x} = \underline{D_x}$
- *P' is i-consistent*

Definition 10 (Maximal right extension)
Let $P = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ be a i-consistent CSP. $P' = (\mathcal{X}, \mathcal{D}', \mathcal{C})$ is a maximal right i consistent extension of D_x for P iff:

- *P' is a right i consistent extension of D_x for P;*
- *$\forall P''$, such that P'' is a right i-consistent extension of D_x for P, $P'' \subset P'$* [6]

3.4 Extrema functions

Let c be an inequality, c_{equ} denotes the equation corresponding to c. More precisely, if c is defined by an expression of the form $f(x_1, \ldots, x_n) \leq 0$ or $f(x_1, \ldots, x_n) \geq 0$, then c_{equ} denotes the equation $f(x_1, \ldots, x_n) = 0$.

[6] A CSP $P = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ is smaller than a CSP $P' = (\mathcal{X}, \mathcal{D}', \mathcal{C})$ if $\mathcal{D} \subseteq \mathcal{D}'$. $\mathcal{D} \subseteq \mathcal{D}'$ means $D_{x_i} \subseteq D_{x'_i}$ for all $i \in 1..n$.

$F_c^{min(x)}(\mathcal{D})$ is an optimal extremum function of c_{equ} for variable x if $F_c^{min(x)}(\mathcal{D})$ computes the smallest value of x which is a solution of c_{equ} [7] in the space delimited by $\mathcal{D}$.

Definition 11 (Optimal extremum function)
Let $P = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ be a CSP, x a variable of $\mathcal{X}$ and c a constraint of $\mathcal{C}$. The optimal extremum function of constraint c_{equ} for variable x is

$F_c^{min(x)}(\mathcal{D}) = min(\pi_x(\vec{v}) \mid c_{equ}(\vec{v})$ *holds.)*

By convention, $F_c^{min(x)}(\mathcal{D})$ returns $\overline{D_x}$ when c_{equ} has no solution in the space delimited by $\mathcal{D}$.

Definition 12 (Extremum function approximation)

Let $P = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ be a CSP, c a constraint of $\mathcal{C}$ and $F_c^{min(x)}(\mathcal{D})$ an optimal extremum function of c_{equ} for variable x.

$AF_c^{min(x)}(\mathcal{D})$ is a safe approximation of $F_c^{min(x)}(\mathcal{D})$ iff: $AF_c^{min(x)}(\mathcal{D}) < F_c^{min(x)}(\mathcal{D})$

3.5 Computing an i–consistent right extension of D_x for P

To define the right i-consistent extension of D_x for a CSP $P = (\mathcal{X}, \mathcal{D}, \mathcal{C})$, we introduce two specific domains, named $\mathcal{D}_{max}$ and $\mathcal{D}_{ext}$:

- $\mathcal{D}_{max}$ is the set of initial domains where D_x has been set to $[\overline{D_x}, \infty)$, i.e., $\mathcal{D}_{max} = \mathcal{D}_{D_x \leftarrow [\overline{D_x}, \infty)}$
- $\mathcal{D}_{ext}$ is the set of initial domains where D_x has been extended to the value of the extremum function of constraint c on x,
  i.e., $\mathcal{D}_{ext} = \mathcal{D}_{D_x \leftarrow [\underline{D_x}, F_c^{min(x)}(\mathcal{D}_{max})]}$

Next proposition defines the right i–consistent extension of D_x for a CSP P with only one single constraint.

Proposition 1 *Let $P = (\mathcal{X}, \mathcal{D}, \{c\})$ be an i-consistent CSP with only one inequality constraint c and let x be a variable of $\mathcal{X}$. Then, $P = (\mathcal{X}, \mathcal{D}_{ext}, \{c\})$ is a maximal right i-consistent extension of D_x for P.*

Proof:
We assume that c is not a tautology [8] which is true for any value of x. So it results from the definition of the extrema functions (definition 9) that we have either:

$\forall \vec{v} \in \mathcal{D}_{ext}\ :\ c(\vec{v})$

or $\forall \vec{v} \in \mathcal{D}_{ext}\ :\ not(c(\vec{v})))$

Since P is i consistent, it results that $\forall \vec{v} \in \mathcal{D}_{ext}\ :\ c(\vec{v})$.

Proposition 2 *Let $P = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ be a i consistent CSP. Let $P' = (\mathcal{X}, \mathcal{D}', \mathcal{C})$ such that:*

- $\forall D_i \in \mathcal{D} \setminus \{D_x\} : D_i = D'_i$

[7] Of course, when c_{equ} is not defined in a subpart of $\mathcal{D}$, $F_c^{min(x)}(\mathcal{D})$ returns a value that is strictly smaller than the smallest value of D_x for which c_{equ} is not defined.

[8] Tautologies can be removed in a pre-processing step.

- $D'_x = [\underline{D_x}, min_{c \in C}(F_c^{min(x)}(\mathcal{D}_{max})))$

Then, P' is a right i-consistent extension on x of P. Furthermore, P' is a maximal right i-consistent extension of P on x.

Proof: From proposition 1, it results that $\forall v \in D', \forall c \in C \; : \; c(v)$. Since P is a conjunction of constraints, P' is i-consistent.
Let $P'' = (\mathcal{X}, \mathcal{D}'', \mathcal{C})$ be an i-consistent extension of D_x for $P = (\mathcal{X}, \mathcal{D}, \mathcal{C})$. Assume that $P' \subset P''$ and that:

- $\forall D''_i \in \mathcal{D}'' \setminus \{D''_x\} : D'_i = D''_i$
- $D'_x \subset D''_x$
- $\underline{D''_x} = \underline{D'_x} = \underline{D_x}$

Assume that c^k is the constraint such that $\overline{D'_x} = F_{c^k}^{min(x)}(\mathcal{D}_{max})$. So, there exists $\vec{v} \in \vec{\mathcal{D}}_{max}$ such that c^k is false, and that $\underline{D'_x} < \Pi_x(\vec{v}) \leq \overline{D''_x}$. Thus, P'' is not i-consistent.

The algorithm in Figure 4 directly follows from property 2. Note that this algorithm is much simpler than the framework introduced by Sam-Haroud and Faltings [9] to compute a local consistency. Both algorithms select relevant extrema from all extrema including intersections between several curves and intersections between curves and interval extremities. However, in our case, the relevant extrema is simply the left most one since we start from an initial i-consistent box (so we know which portion of the space is a solution) and we extend only one variable domain to the right. This algorithm only searches for the left, most extrema, thus, it is linear if the extrema functions can be computed in constant time. The next section shows that the left most extrema can be computed very efficiently.

```
function i-extension(x, D_max, C): real
  I ← Max-Value
  for c in C
      I ← Min(I, F_c^min(x)(D_max))
  return I
  End function
```

Figure 4: function i-extension

3.6 Computing extrema functions

Optimal extrema functions for variable x of constraint c can trivially be computed if c is either a monotonic on x, or if D_x can be decomposed in subdomains where c is monotonic on x. Such constraints are usually called primitive constraints[3]. The set of primitive constraints is infinite and includes the following constraints: $\{x = y, x \leq y, x < y, x \neq y, z = x + y, z = x * y, x = -y, y = sin(x), y = cos(x), y = e^x, y = abs(x), z = x^y, \ldots\}$.

Example 3 *The constraint $x^3 = y$ is primitive: the right extrema function for x is :*

$$F_x^{min}(D_x, D_y) = max(\underline{D_x}, \sqrt[3]{\underline{D_y}}))$$

For a non-primitive constraint c, we will approximate the e-consistent box for c_{equ} in the space delimited by domains $\mathcal{D}_{D_x \leftarrow D_{max}}$. The methods introduced to compute Box-consistency provide an efficient way to compute such a safe approximation of $F_c^{min(x)}(\mathcal{D}_{max})$. The key observation is that extrema functions are *univariate functions* which can be tackled by the *Newton method* implemented in the Box-consistency.

So, consider the i-consistent extension of D_x for CSP P = *(A\V,C)* and an inequality c $\in$ *C*, To compute a safe approximation of the extrema functions for x of constraint c, we could just compute a Box-consistent interval for x with regard to c_{equ}. Box-consistency would yield an interval D'_x such that $\Pi_x(sol(c_{equ}, \mathcal{D}_{max})) \subset D'_x$. Thus, $\underline{D_x} = AF_c^{min(x)}(\mathcal{D}_{max})$.

As a matter of fact, a complete computation of Box-consistency is not required. The LNAR procedure [II] used in Box-consistency finds the left most zero of the interval extension of the univariate function on x derived from c_{equ} by replacing all variables but x by their domains. Of course, when the function i-extension (see. fig. 4) uses approximations of extrema functions, the i-extension of the domain of x may not be maximal.

4 A ballistic example

In this section, we give a small ballistic application which illustrates the capabilities of our system. The problem consists of finding the maximum mechanical tolerances when an object is launched in a uniform gravitational field $\vec{g}$, with an initial speed $\vec{V_i}$ which has an incidence α with the ground (see fig. 5).

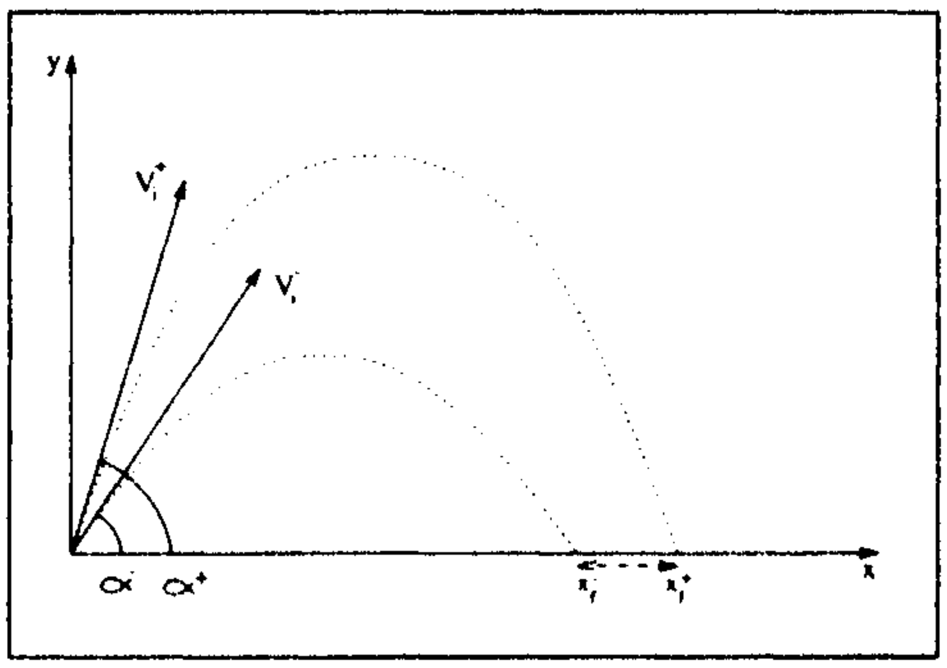


Figure 5: Possible trajectories of the projectile

The strong requirement is that the object must fall inside a predefined interval.

4.1 Modeling of the problem

The initial speed and incidence of the bullet can be stated as follows:

$$x_f = V_x t_f \qquad y_f = -\tfrac{1}{2} g t_f^2 + V_y t_f$$
$$V_x = V\cos(\alpha) \qquad V_y = V\sin(\alpha)$$

These equations give the impact point (x_f, y_f) of the bullet when $t = t_f$ and $y_f = 0$: $t = V_y \frac{2}{g}$
Thus, $x = V_x \frac{2V_y}{g} = \frac{2V^2}{g} \cos(\alpha)\sin(\alpha)$

4.2 Computing i-consistency extension of D_α

The target is defined by the interval [220, 250]. Now, assume that the bullet falls on the target when $\alpha \in$ [32,35] and $V \in$ [49.2,50.1]. So the initial i consistent box is defined by :
$\forall \alpha \in D_\alpha$:

$$\frac{2}{9.81} * [49.2, 50.1]^2 * cos(\alpha) * sin(\alpha) \geq 220$$
$$\frac{2}{9.81} * [49.2, 50.1]^2 * cos(\alpha) * sin(\alpha) \leq 250 \quad \text{(O}$$

To extend D_α to the right by iconsistency, we have to check whether the box at the right of the i consistent box is i -consistent. Thus, we have to find the left most bound of D_α for c_{equ} and c'_{equ} with $D_{max} = [35, 90]$. To find these bounds, we have used Numerica [10] to compute Box consistent intervals for c_{equ} and c'_{equ}. The left most bound of these intervals respectively are 58.4 and 38.4. So, D_α can be extended by iconsistency to interval [32,38.4] (see Fig. 6).

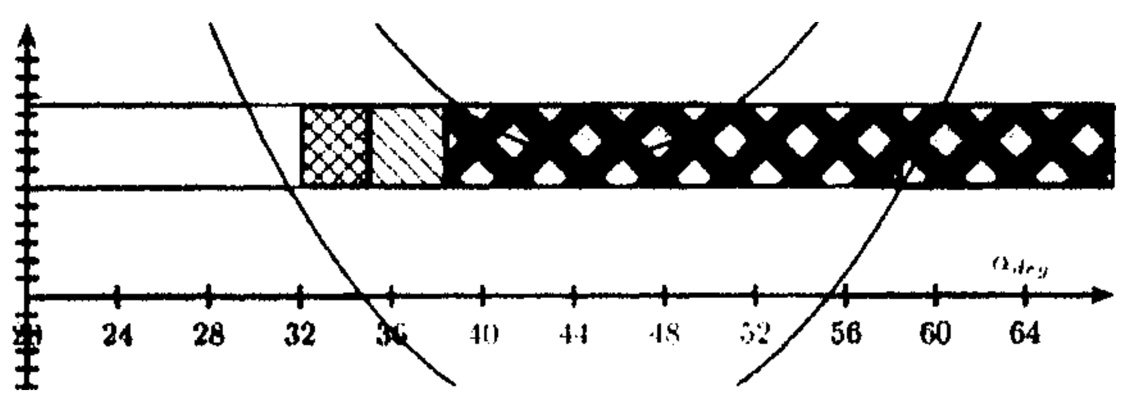


Figure 6: The ballistic constraints and boxes

Now, consider the three dimensional version of this problem where the target is defined by a rectangular area of space $R_x \times R_y \times R_z$ such that $220 \leq R_x \leq 250$, $R_y = 0$ and $-50 \leq R_z \leq 50$. Let β be the angle between x and the projection of $\vec{Y}$ on the plane defined by $y = 0$. We know that the box defined by $D_\alpha = [34, 35], D_V$ — [49.2,50.1] and D_B = [0,0] is i consistent.
To extend *ft* by i consistency, we have computed with Numerica the Box-consistent intervals for the equations derived from the following inequalities :

$$220 \leq \frac{2}{9.81} * V^2 * sin(\alpha) * cos(\alpha) * cos(\beta)$$
$$250 \geq \frac{2}{9.81} * V^2 * sin(\alpha) * cos(\alpha) * cos(\beta)$$
$$-50 \leq \frac{2}{9.81} * V^2 * sin(\alpha) * cos(\alpha) * sin(\beta)$$
$$50 \geq \frac{2}{9.81} * V^2 * sin(\alpha) * cos(\alpha) * sin(\beta)$$

The left most bound of these intervals is 1.4; thus, D_β can be extended by i-consistency to the interval [0,1.4].

5 Conclusion

Tins paper has introduced an effective framework for extending the domain of one variable in an already consistent CSP. Extending the domain of one variable is a critical issue in applications where the tolerance of a. component determines its cost.

Contrary to Ward et al [13] we do not impose any restrictions on the form of the constraints. The approach suggested by Sam-Haroud and Faltings [9] is more general since they do not know an initial solution but its computation cost is very high. The key point of our framework is the definition of univariate extrema functions which can be computed efficiently.
An interesting way to explore concerns maximizing the size (or volume) of iconsistent boxes.

Acknowledgements

Thanks to Gilles Trombettoni for his careful reading and helpful comments on earlier drafts of this paper. Thanks also to Olivier Lhomme, Jean-Paul Stromboni and Alexander Semenov for interesting suggestions.

References

[I] F. Benharnou, I). Mc A Hester and P. Van Hentenryck. CLP(intervais) revisited. In *Proc. 1LPS.* MIT Press, 1994.

[2] F. Benhamou and W. Older. Applying interval arithmetic to real, integer and boolean constraints. *Journal of Logic Programming,* 1997.

[3] H. Collavizza, F. Delohel and M Riieher. A note on partial consistencies over continuous domains. In *Proc. of CP 98,* LNCS 1520, Springer Verlag, 1998.

[4] E. Davis. Constraint propagation with interval labels. *Artificial Intelligence,* 32:281 -331, 1987.

[5] B. Faltings. Are-consistency for continuous variables. *Artificial Intelligence,* 65:363 376, 1994.

[6] B. Faltings and E. (telle. Local consistency for ternary numeric constraints. *IJCAI'91,* pages 392 397, 1997.

[7] O. Lhomme. Consistency techniques for numeric csp. *Proc. IJCAI93,* pp. 232 238, 1993.

[8] A. Mackworth. Consistency in networks of relations. *Artificial Intelligence,* 8(I):pp. 99 118, 1977.

[9] D.J Sam-Haroud and B. Faltings. Consistency techniques for continuous constraints. *Constraints,* I(I& 2):85 118, September 1996.

[10] P. Van Hentenryck, V. Deville and L. Michel. *Numerica. A modeling language for global optimization.* MIT Press, 1997.

[II] L). Kapur, D. Mc A Hester and P. Van Hentenryck. *Solving Polynomial Systems Using a Branch, and Prune Approach SI AM Journal on Numerical Analysis,* 34(2), April 1997.

[12] D. Waltz. *The Psychology of Computer Vision,* chap. Understanding line drawing of scenes with shadow. McGraw-Hill, 1975.

[13] A.C Ward, T. Lozano-Perez and W.P. Seering. Extending the constraint propagation of intervals. In *Proc. of the llth IJCAl,* pages 1453-1458, 1989.