

Local consistency for ternary numeric constraints

Boi Faltings and Esther Gelle
Artificial Intelligence Laboratory (LIA)
Swiss Federal Institute of Technology (EPFL)
IN-Ecublens, 1015 Lausanne, Switzerland
faltings/gelle@lia.di.epfl.ch

Abstract

We consider algorithms of the Waltz type for computing local consistency (also called arc-consistency) for constraints over numeric domains. Many commonly used propagation rules do not in fact enforce local consistency. We extend the propagation rule given by Faltings [Faltings, 1994] to the case of ternary constraints. Since any general n -ary continuous constraint can be represented as a collection of ternary ones, this also covers n -ary constraints in general. We show how the propagation can be implemented efficiently. The new algorithm gives significantly tighter labellings than previous propagation algorithms on most problems.

1 Introduction

The constraint satisfaction problem (CSP) is ubiquitous in many practical applications. We consider in particular CSPs with continuous-valued variables, given by:

- a set of variables $\{X_i\}$.
- a set of ternary constraints C_{X_i, X_j, X_k} , defined as inequalities $f(x_i, x_j, x_k) \geq 0$. Note that an equality $f = 0$ can be modelled as a conjunction of two inequalities $f \geq 0$ and $f \leq 0$. Binary and unary constraints are defined similarly with the corresponding number of variables.

A *solution* to a CSP consists of value assignments $\{X_i = x_i\}$ to all variables such that all constraints are satisfied. Throughout this paper, we use upper case letters for variables and lower case letters for particular values.

For linear constraints, there are efficient optimization algorithms for finding *single* solutions to CSPs. For non-linear constraints, optimization algorithms only apply to special cases. More importantly, there are many applications in configuration, diagnosis or planning where it is important to know not a *single* solution, but the space of *all* possible solutions. Algorithms for determining the space of all solutions also find applications in numerical analysis ([Benhamou et al., 1994]). A solution space can be represented by a set of *labels* defining the consistent values or value combinations. In numeric domains,

it is very difficult to represent sets of value combinations. However, sets of values for individual variables can be represented efficiently by collections of *intervals*. We consider algorithms for computing labellings which are *locally consistent* according to the following definition:

Definition 1 A labelling is locally consistent if for every X_i, X_j and every value $x_i \in L_i$, there exists a value $x_j \in L_j$ such that for every constraint C_{X_i, X_j, X_k} involving X_i and X_j and possibly another variable X_k , there exists a value $x_k \in L_k$ such that C_{X_i, X_j, X_k} is satisfied for $X_i = x_i, X_j = x_j, X_k = x_k$:

$$(\forall i, j) \quad (\forall x_i \in L_i)(\exists x_j \in L_j) \\ (\forall C_{X_i, X_j, X_k})(\exists x_k \in L_k) C_{X_i, X_j, X_k}(x_i, x_j, x_k)$$

Note that this definition is different from others (e.g., [Davis, 1987]) in that it requires that one single value x_j can satisfy *all* constraints. This ensures that the locally consistent labelling of the ternary system is at least as tight as the locally consistent labelling of its binary projection. Note that our work distinguishes itself from others, for example [Hyvönen, 1992] or [Lhomme, 1993], in that we are only considering efficient and general algorithms for *local* consistency, without any attempt to impose global consistency.

2 Local consistency with numeric constraints

A locally consistent labelling can be computed using the Waltz algorithm. It consists of applying a *propagation rule* to the labels of every variable pair until it results in no more change. While local consistency is very often applied for CSP with discrete domains, in continuous domains Davis ([Davis, 1987]) has reported a number of very negative results, including the fact that the Waltz algorithm does not guarantee a locally consistent labelling and may often fail to terminate. [Faltings, 1994] has shown that many of his negative results are due only to an overly straightforward formulation of the propagation rule used in the Waltz algorithm:

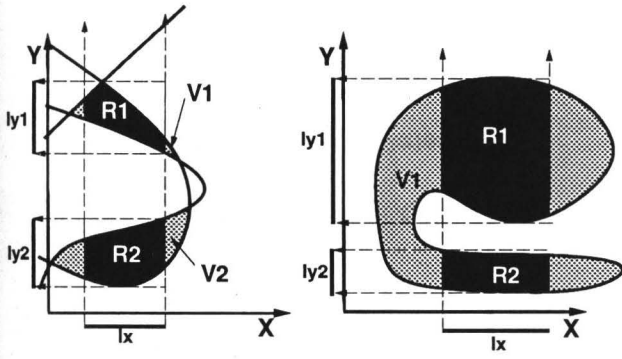


Figure 1: Two examples of total constraints. The constraint on the left consists of the two feasible regions $V1$ and $V2$. When propagating from X to Y , the interval I_x generates the restricted regions $R1$ and $R2$ which project into interval I_{y1} and I_{y2} . The example on the right shows that multiple restricted regions $R1$ and $R2$ can result from a single consistent region $V1$.

- Davis assumes that only interval boundaries are propagated, but with nonlinear constraints local extrema and intersections between constraints may also become interval bounds.
- Davis assumes that each constraint between the same set of variables is propagated individually, thus creating implicit cycles in the constraint network when several constraints involve the same variables.

The shortcomings introduced by these assumptions are already apparent in the following simple example of a parabola and a sphere:

$$X_1^2 + 1/2 * X_2 + 2 * (X_3 - 6) \geq 0$$

$$X_1^2 + X_3^2 + X_4^2 - 25 \leq 0$$

with initial intervals of X_i $i = 1, 2, 3, 4$ of $[-10, 10]$. Here, Davis' propagation rule results in no refinement whatsoever of the initial labels. However, upon closer consideration, we find that:

- X_3 can never become greater than 5. This bound is not found since it is reached at a local extremum of a constraint.
- X_3 can never become smaller than -3.36. This bound is not found since it is given by the intersection of the two constraints.

A more powerful propagation rule can be obtained by generalizing the propagation rules used in discrete domains to the infinite, continuous case. To do this, it is necessary to consider all constraints involving the same pair of variables X and Y simultaneously rather than individually; we call their conjunction the *total constraint* between X and Y . It may consist of several regions $V_1, \dots, V_k$ of feasible value combinations (Figure 1).

In a propagation step from X to Y , the label of Y is modified to exclude those values which are not compatible with any value in the label of X . This can be

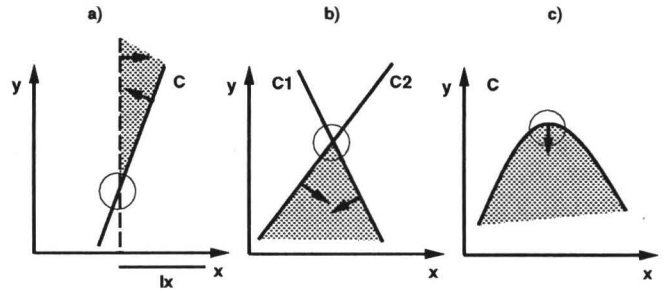


Figure 2: The three different types of local extrema: a) a minimum a) and two maxima b) and c).

accomplished by first restricting the feasible regions of the total constraint to those within the label of X , called the *restricted regions* (Figure 1) $R_1, \dots, R_l$. Projecting these restricted regions onto the Y -axis gives the intervals of Y -values admitted by the propagation. [Faltings, 1994] shows that this propagation rule guarantees a locally consistent labelling upon termination.

Implementing such a rule poses some difficulties, since the regions making up the total constraint can become quite complex (see Figure 1). [Faltings, 1994] gives a method for propagation without explicitly considering the regions formed by the constraints. It is based on propagating *extrema* of boundaries of restricted regions. There are three types of extrema (see Figure 2):

- intersections between constraints and interval bounds
- constraint intersections
- local extrema of constraint curves

We consider the propagation of the label of X to that of Y through a restricted region of a constraint between X and Y . We let $B(R)$ be the continuous boundary of a restricted region R . Extrema are then the local maxima and minima in Y of $B(R)$ with respect to X . Note in particular that extrema here refer to the boundary $B(R)$, not the region R itself. As a notation, we let $\max_Y(B(R), y_0)$ be true if and only if $B(R)$ has a local maximum in Y with Y -coordinate y_0 , and $\min_Y(B(R), y_0)$ if it has a local minimum. Note that if R is bounded by a set of constraint curves, the extrema of its boundary $B(R)$ are a subset of the extrema and intersections of these curves. Only those which satisfy all other constraints bounding R are valid, the others are ignored. Thus, the set of extrema can be found by purely local considerations.

We now define the function $\alpha_Y(R, y)$ to be the difference in the number of maxima and the number of minima at y -coordinates greater than or equal to y :

$$\alpha_Y(R, y) = |\{y_0 | \max_Y(B(R), y_0) \wedge y_0 > y\}| - |\{y_0 | \min_Y(B(R), y_0) \wedge y_0 > y\}|$$

It is then possible to prove ([Faltings, 1994]) that:

- if there is a point $(x^*, y^*) \in R$, $\alpha_Y(R, y^*) > 0$, (Lemma 2.9) and

- if there is no point $(x^*, y^*) \in R$, $\alpha_Y(R, y^*) = 0$ (Lemma 2.7)

The propagation rule should eliminate a value y^* from the label L_Y if and only if it is in no region, i.e. $\alpha_Y(R_i, y^*) = 0$ for all R_i . Since $\alpha_Y(R_i, \dots) \geq 0$ for any R_i , this can be conveniently expressed as $\sum_i \alpha_Y(R_i, y^*) = 0$. By rewriting this as:

$$\sum_i \alpha_Y(R_i, y^*) = \sum_i |\{y_0 | \max_Y(B(R_i), y_0) \wedge y_0 > y^*\}| - \sum_i |\{y_0 | \min_Y(B(R_i), y_0) \wedge y_0 > y^*\}|$$

we only require the total number of extrema in *all* regions taken together, without any consideration of the regions they belong to. The set of illegal y^* , and conversely the set of legal y , can thus be characterized without knowing which extrema belong to which region!

The following propagation rule computes the maximal intervals of admissible y and thus achieves local consistency (for the proof, see [Faltings, 1994]):

1. $I_y \leftarrow \{\}$
2. compute the set *MAX* of local maxima and the set *MIN* of local minima of all restricted regions $r \in R(Q, I_x)$, $Q \in C^t(x, y)$.
3. Filter both *MAX* and *MIN* to keep only those extrema which satisfy all other constraints, i.e. actually bound a legal region. Order the two sets of points according to their y -coordinate (these sets are obtained directly by consideration of individual constraints $c(x, y)$).
4. set $index \leftarrow 0$, consider the extrema e in *MIN* and *MAX* in decreasing order of their y -coordinate, where elements of *MAX* are always considered before elements of *MIN*, and do:
 - (a) if $e \in MAX$, $index \leftarrow index + 1$. If $e \in MIN$, $index \leftarrow index - 1$.
 - (b) if $index$ has just changed from 0 to 1, set $y_{lower} \leftarrow y - coordinate(e)$.
 - (c) if $index$ has just changed from 1 to 0, set $y_{upper} \leftarrow y - coordinate(e)$ and add the interval $[y_{lower}..y_{upper}]$ to I_y .
5. return I_y .

We assume that all local extrema of constraints as well as constraint intersections have been precomputed before running the propagation, so that the computation of *MAX* and *MIN* in step 2 only requires evaluating the constraints at interval bounds; the amount of constraint manipulation required in each propagation step is exactly the same as that of other more straightforward propagation rules based only on interval bounds.

3 Generalization to ternary constraints

In practice, constraint networks are almost never binary. The main novel contribution of this paper is to generalize the propagation rule of [Faltings, 1994] to ternary

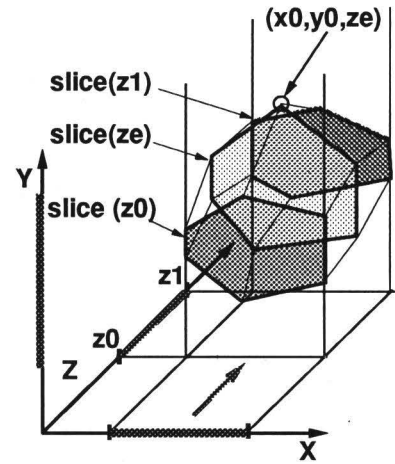


Figure 3: Propagating intervals through a set of ternary constraints. The point (x_0, y_0, z_e) is an extremum in Y of the ternary constraint.

constraints over continuous variables. The generalization again makes use of the property that the propagation rule only needs to know the complete set of extrema of all feasible regions, but not what regions there are.

Propagation with a single third variable Z We first consider the case of propagating the label of X to Y through a set of constraints C_{XYZ} involving a single third variable Z (see Figure 3). The consistency condition of Definition 1 for the label of Y is now:

$$(\forall y \in L_Y)(\exists x \in L_X)(\exists z \in L_Z)C_{XYZ}(x, y, z)$$

We define the *projection* of the ternary constraint:

$$P_{XY}(C_{XYZ}) = \{(x, y) | (\exists z \in L_Z)C_{XYZ}(x, y, z)\}$$

so that the local consistency condition for L_Y for the ternary constraint C_{XYZ} becomes the consistency condition for the binary constraint $P_{XY}(C_{XYZ})$. Thus, propagation through the projection will give a consistent labelling.

In order to apply the propagation rule, we need to determine the extrema in the boundaries of this projection. The following lemma indicates where to find them:

Lemma 1 Assume that (x_e, y_e) is an extremum of $P_{XY}(C_{XYZ})$. Then, there must exist a $z_e \in L_Z$ such that (x_e, y_e, z_e) is an extremum of C_{XYZ} restricted to values of Z within its label L_Z .

Proof:

Since (x_e, y_e) is in $P_{XY}(C_{XYZ})$, there must be a z_e such that (x_e, y_e, z_e) is in C_{XYZ} . On the other hand, since (x_e, y_e) is an extremum in y of $P_{XY}(C_{XYZ})$, there is a small interval of x' around x_e such that there is no point $(x_e, y') \in P_{XY}(C_{XYZ})$ with $y' > y_e$ (maximum) or $y' < y_e$ (minimum). But this means that there is no z' such that $(x', y', z') \in C_{XYZ}$, which is just the condition for (x_e, y_e, z_e) to be an extremum of the ternary constraint. QED.

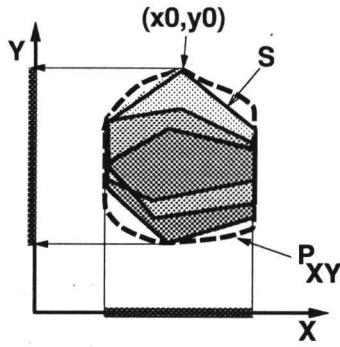


Figure 4: The three slices generate the set S which mirrors all extrema of P_{XY} even though it does not necessarily cover P_{XY} (shown by dashed lines) exactly.

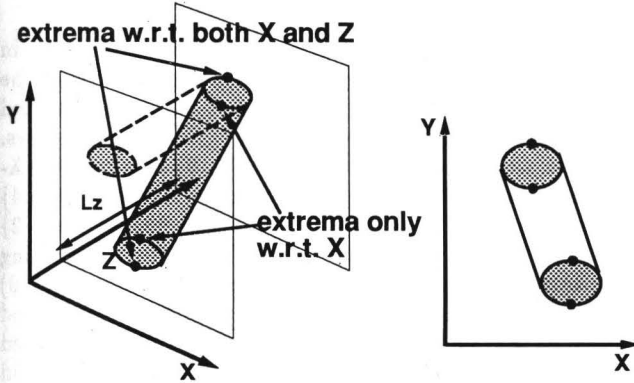


Figure 5: The figure on the left defines a cylinder rising in Z . The two slices leave a gap which is not present in P_{XY} . The projection of the cylinder on axes X and Y with the slices hatched is shown on the right.

Extrema of ternary constraints can fall on interval bounds, local extrema, constraint intersections, but also combinations thereof. For example, the intersection of two constraints involving the same variables forms a curve which can have an extremum, or a constraint curve can have a local extremum in one coordinate and reach interval bound in the other. Let $Z = \{z_{e1}, z_{e2}, \dots, z_{ek}\}$ be the set of Z -coordinates of all extrema. Each z_{ei} defines a "cut" through the space spanned by all constraints involving X and Y , called a *slice* (Figure 3). Now consider the set S :

$$S = \bigcup_{z_{ej} \in Z} \{(x, y) | C_{XYZ}(x, y, z_{ej})\}$$

which is the union of all slices at extrema of Z (Figure 4). The set S does not necessarily cover $P_{XY}(C_{XYZ})$, but because of Lemma 1 it contains all extrema. However, it is not true that the union of all extrema of slices in S will give the correct propagation, since it also contains *interior* extrema which would lead to false gaps during propagation. For example, the propagation shown in Figure 5 generates disjoint slices even though the pro-

jection should be one continuous region.

These interior extrema are characterized by the fact that they are not also extrema of C_{XYZ} with respect to Z . Thus, the set of extrema E of $P_{XY}(C_{XYZ})$ is the set of all extrema of some slice which are also extrema with respect to Z . We can show the following:

Lemma 2 *Updating the label of Y by propagating L_X through E will give the same result as propagating it through $P_{XY}(C_{XYZ})$, and thus through C_{XYZ} itself.*

Proof:

Interval bounds of $P_Y(C_{XYZ})$ can only fall on extrema of $P_{XY}(C_{XYZ})$ and thus of C_{XYZ} itself. By construction, E contains all extrema in Y of $P_{XY}(C_{XYZ})$, with the interior ones filtered out. Thus, the projection of S onto the Y -axis will be identical to that of $P_{XY}(C_{XYZ})$.

QED

and we have a correct propagation rule for a set of constraints with a single third variable.

Propagating constraints with several third variables When there are several third variables $Z_1, Z_2, \dots, Z_k$ with constraints C_{XYZ_i} , labels must be propagated through their *intersection*. Since the third variables are all distinct, the projection of their intersection is the intersection of the projections, i.e.:

$$\bigcap_{i=1..k} P_{XY}(C_{XYZ_i})$$

This intersection can be carried out implicitly by (i) filtering all extrema (x_e, y_e) such that for some third variable Z_i , there does not exist a z_{ei} such that (x_e, y_e, z_{ei}) satisfies C_{XYZ_i} , and (ii) adding all extrema resulting from intersections of C_{XYZ_i} and C_{XYZ_j} .

Identifying slices Slices need to identify all points which could possibly give rise to extrema of the projection P_{X_i, X_j} of a constraint. Consider thus a total constraint $C_{X_i, X_j}^t = \{C_{X_i, X_j, Z_1}^1, \dots, C_{X_i, X_j, Z_k}^k\}$ involving k other variables $Z_1..Z_k$.

The first set of slices to be considered are those which fall at *interval bounds* of some Z_i . The second set of slices to be considered are *local extrema* of individual constraint curves. Consider a single ternary constraint C_{XYZ_i} . It has a local extremum in Y at (x_e, y_e, z_e) if both:

- either $\delta C / \delta X(x_e, y_e, z_e) = 0$ or x_e is an interval bound
- either $\delta C / \delta Z(x_e, y_e, z_e) = 0$ or z_e is an interval bound

The set of relevant slices is obtained by considering all constraints which involve X and Y and collecting all z_e satisfying the above conditions and falling into the label of Z_i, L_{Z_i} .

The third set of slices is generated by *intersections* between constraints. Here, the situation can be quite

The is the case exactly if the normal vectors of both constraints lie in a plane that also includes the Y -axis, i.e. $n(\vec{C}_1) + \alpha n(\vec{C}_2)$ is a vector parallel to the Y axis, for some α . This leads to a system of equations for such extrema which can be solved using numerical methods.

The binary propagation rule is thus adapted to ternary constraints by replacing step 2) as follows:

2. $MIN \leftarrow \{\}, MAX \leftarrow \{\}$
 $Z \leftarrow \{Z_i | \exists C_{XYZ_i}\}$
 for all $Z_i \in Z$ do
 $S \leftarrow$ all slices of Z_i
 for every slice $(z_{e1}, z_{e2} \dots) \in S$ do
 1. add to MAX all local maxima and to MIN all local minima of all restricted regions R_i in the slice $C^t(x, y, z_{ei})$, obtained directly by consideration of individual inequalities and their intersections.
 2. filter from MAX and MIN all those extrema which are not also extrema with respect to Z_i .
 3. for all other constraints $C_{XYZ_k}, Z_k \in Z$ filter from MAX and MIN all those extrema which are not satisfiable by the constraint C_{XYZ_k} , and add all extrema which correspond to intersections between a constraint C_{XYZ_i} and C_{XYZ_k} .

The following theorem shows that using this propagation rule in the Waltz algorithm results in a locally consistent labelling:

Theorem 1 *The ternary propagation rule reaches quiescence only when the labels are locally consistent.*

Proof:

Follows from the correctness of the binary propagation rule (Theorem 2.1 in [Faltings, 1994]) and the equivalence of propagating through S and P_{XY} .

QED

Example of a propagation step. We consider the propagation of X_1 to X_3 in the system of constraints already mentioned in section 2:

$$X_1^2 + 1/2 * X_2 + 2 * (X_3 - 6) \geq 0$$

$$X_1^2 + X_3^2 + X_4^2 - 25 \leq 0$$

with four variables $X_1, \dots, X_4$ and initial labels of $[-10, 10]$ for X_1 to X_4 . The total constraint is thus formed by $\{C_1, C_2\}$. C_2 has a maximum in X_3 at $\{X_1 = 0, X_3 = 5, X_4 = 0\}$ and a minimum in X_3 at $\{X_1 = 0, X_3 = -5, X_4 = 0\}$. The projections on X_2 and X_4 of combinations of extrema (individual extrema and interval bounds) in X_3 give rise to the following potential slices:

$$X_2 \in \{-10, 10\} \quad \text{and} \quad X_4 \in \{-10, 0, 10\}$$

For each of the slices defined by substituting combinations of these values in the constraints, the label of X_1 , $[-10, 10]$, is propagated to X_3 . Only the slices $(X_2, X_4) = (-10, 0)$ and $(10, 0)$ give rise to valid extrema

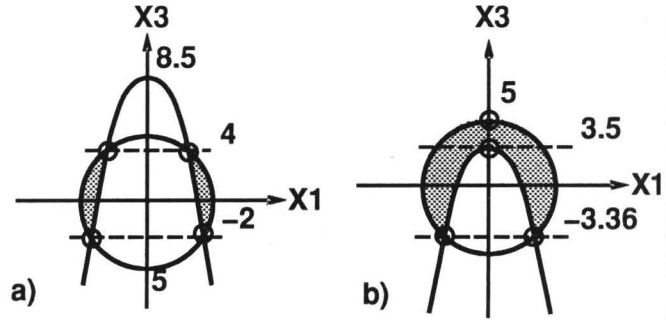


Figure 6: The propagation of $X_1 = [-10, 10]$ through a) the slice at $\{X_2 = -10, X_4 = 0\}$ and b) through the slice at $\{X_2 = 10, X_4 = 0\}$ produces two maxima and two minima for each slice.

(i.e. satisfying all other constraints); they are shown in Figure 6. The result propagating X_1 to X_3 through the total constraint formed by C_1 and C_2 is then obtained by collecting the valid extrema produced by these slices. The first slice with $(X_2, X_4) = (-10, 0)$ has two MAXIMA in X_3 at $\{X_1 = -3, X_3 = 4\}$ and $\{X_1 = 3, X_3 = 4\}$ and two MINIMA in X_3 at $\{X_1 = -4.583, X_3 = -2\}$ and $\{X_1 = 4.583, X_3 = -2\}$ all of them resulting from intersections. The second slice $(X_2, X_4) = (10, 0)$ has two local MAXIMA at $\{X_1 = 0, X_3 = 3.5\}$ and $\{X_1 = 0, X_3 = 5\}$ as well as two MINIMA produced by intersections at $\{X_1 = -3.7, X_3 = -3.359\}$ and $\{X_1 = 3.7, X_3 = -3.359\}$. Ordering all extrema in X_3 and applying step 4 of the propagation rule gives the single consistent interval $[-3.359, 5]$. In this example, the Davis method does not alter the labels $[-10..10]$ for all variables at all, whereas our propagation rule gives $X_1 = [-5, 5]$, $X_2 = [-10, 10]$, $X_3 = [-3.359, 5]$ and $X_4 = [-5, 5]$.

4 Performance on random examples

In this section, we would like to compare our algorithm with the propagation of individual constraints ([Davis, 1987]) and show on randomly generated examples why our algorithm is better than Davis'. Numeric values in these examples are all rounded on 2 decimals.

Supposing that the restricted region always falls within the propagated intervals, we applied Davis propagation to the following examples:

Constraint set 1

- C_1 $-4 * X_5 - X_6 + 6 * X_9 + 6 > 0$
- C_2 $4 * X_4 - 9 * X_5 - 5 * X_8 - 5 > 0$
- C_3 $-9 * X_2 + 7 * X_3 + 4 * X_5 + 4 > 0$
- C_4 $-X_6 + 10 * X_9 - 4 * X_{10} - 4 > 0$
- C_5 $X_4 + X_7 - X_8 - 1 > 0$
- C_6 $-4 * X_4 + X_7 - X_8 + 1 > 0$
- C_7 $-6 * X_3 - 2 * X_5 + 6 * X_6 + 6 > 0$
- C_8 $6 * X_2 - X_4 + 7 * X_5 + 7 > 0$
- C_9 $8 * X_1 - 5 * X_2 - 8 * X_3 - 8 > 0$

X_i	Local C.	Davis	Simplex
X_1	[-10, 10]	[-10, 10]	[-10, 10]
X_2	[-10, 7.2]	[-10, 10]	[-8, 85, 7.2]
X_3	[-10, 10]	[-10, 10]	[-10, 10]
X_4	[-10, 5.25]	[-10, 5.25]	[-10, 5.25]
X_5	[-6.86, 7.33]	[-10, 7.33]	[-5.35, 7.33]
X_6	[-10, 10]	[-10, 10]	[-9.84, 10]
X_7	[-9.4, 10]	[-10, 10]	[-9.4, 10]
X_8	[-10, 9.4]	[-10, 10]	[-10, 7.16]
X_9	[-4.6, 10]	[-4.6, 10]	[-3.41, 10]
X_{10}	[-10, 10]	[-10, 10]	[-10, 10]

Table 1: Solutions for constraint set 1.

Constraint set 2

$$\begin{aligned}
C_1 \quad & -16 + \frac{7}{10} * X_9 * X_5 + \frac{7}{5} * X_6 - \frac{6}{5} * X_5 - \frac{6}{5} * X_9 - \\
& \frac{13}{50} * X_9 * X_6 - \frac{13}{50} * X_5 * X_6 + \frac{1}{10} * X_6 * X_6 + \\
& \frac{3}{5} * X_5 * X_5 + \frac{3}{5} * X_9 * X_9 < 0 \\
C_2 \quad & 4 * X_4 - 9 * X_5 - 5 * X_8 - 5 > 0 \\
C_3 \quad & -9 * X_2 + 7 * X_3 + 4 * X_5 + 4 > 0 \\
C_4 \quad & -\frac{4}{5} * X_{10} - \frac{4}{5} * X_6 + \frac{1}{10} * X_{10} * X_9 + \frac{1}{10} * X_6 * X_9 + \\
& \frac{1}{10} * X_6 * X_{10} + \frac{17}{50} * X_9 + \frac{1}{50} * X_9 * X_9 + \frac{1}{4} * X_6 * X_6 + \\
& \frac{1}{4} * X_{10} * X_{10} - \frac{22}{3} < 0 \\
C_5 \quad & -6 * X_3 - 2 * X_5 + 6 * X_6 + 6 > 0 \\
C_6 \quad & 6 * X_2 - X_4 + 7 * X_5 + 7 > 0 \\
C_7 \quad & 8 * X_1 - 5 * X_2 - 8 * X_3 - 8 > 0
\end{aligned}$$

A comparison of the results of Davis' algorithm and the local consistency rule (Table 1) reveals that local consistency prunes labels more drastically as it considers intersections between constraints defined on the same pairs of variables. X_5 is an example of such a variable involved in several constraints. When compared with the simplex algorithm, which computes the globally optimal solution for constraint set 1, the labels of X_2 , X_5 , X_7 and X_8 show that local consistency is better than the Davis rule although it does not achieve global consistency. In the second example, two nonlinear constraints C_1 and C_4 have been added to a set of linear inequalities. Here, the local consistency algorithm is far better because, additionally to label boundaries, it takes into account the individual extrema of nonlinear constraints (Table 2).

We randomly generated about 300 mixed constraints sets similar to constraint set 2. For each solution we evaluated the average number of labels different from the initial label and how much the labels had been reduced on average. Davis' algorithm reduced 3 out of 10 labels whereas local consistency reduced nearly 7 labels. The amount of pruning was 7.48% for Davis and 25.97 % for local consistency.

5 Conclusions

Among all techniques developed for constraint-based reasoning, local consistency and its weaker variants such as forward checking are by far the most widely used techniques in practice. This is due to the fact that it can be computed efficiently but still provides dramatic pruning of the search space, especially when some variables are already assigned values.

X_i	Local Consistency	Davis
X_1	[-10, 10]	[-10, 10]
X_2	[-10, 7.15]	[-10, 10]
X_3	[-10, 5.32]	[-10, 10]
X_4	[-10, 10]	[-10, 10]
X_5	[-4.6, 7.16]	[-10, 9.44]
X_6	[-4.37, 2.79]	[-10, 10]
X_8	[-10, 10]	[-10, 10]
X_9	[-6.35, 4.24]	[-10, 10]
X_{10}	[-6.91, 10]	[-10, 10]

Table 2: Solutions for constraint set 2.

In continuous domains, straightforward implementations of Waltz propagation does not guarantee useful results ([Davis, 1987]). A modified propagation rule which guarantees local consistency has been given in [Faltings, 1994] for binary constraints. In this paper, we have shown how the rule can be generalized to ternary and thus n-ary constraints. The Waltz algorithm using the new propagation rule gives significantly better results than the commonly used propagation rule analyzed by Davis. This result should allow significant improvements in many applications of local consistency.

References

- [Benhamou et al., 1994] F. Benhamou, D. McAllester, P. Van Hentenryck: "CLP(Intervals) Revisited," Technical Report No. CS-94-18, Université Aix-Marseille, April 1994
- [Davis, 1987] Ernest Davis: "Constraint Propagation with Interval Labels," *Artificial Intelligence* 32, 1987, pp. 281-331
- [Faltings, 1994] Boi Faltings: "Arc-consistency for continuous variables," *Artificial Intelligence* 65, 1994, pp. 363-376
- [Freuder, 1982] Eugene Freuder: "A sufficient condition for backtrack-free search," *Journal of the ACM* 29(1), 1982, pp. 24-32
- [Hyvönen, 1992] E. Hyvönen: "Constraint reasoning based on interval arithmetic: the tolerance propagation approach," *Artificial Intelligence* 58(1992), pp. 71-112
- [Lhomme, 1993] O. Lhomme: "Consistency Techniques for Numeric CSPs," *Proceedings of the 13th IJCAI*, 1993, pp. 232-238
- [Sam-Haroud and Faltings, 1996] Djamila Sam-Haroud, Boi Faltings: "Consistency Techniques for Continuous Constraints," *Constraints* 1(1), 1996, pp. 85-118
- [Waltz, 1975] David Waltz: "Understanding line drawings of scenes with shadows," in P.H. Winston(ed.): *The Psychology of Computer Vision*, McGraw-Hill, NY 1975