Using Constraint Programming for Solving Distance CSP with uncertainty

Student: Carlos Grandon[1,3]
Supervisor: Bertrand Neveu[2,3]

[1] cgrandon@sophia.inria.fr
[2] neveu@sophia.inria.fr
[3] COPRIN project I3S-CNRS/INRIA/CERMICS
INRIA, 2004 Route des Lucioles, BP 93
06902 Sophia-Antipolis, France

Abstract. Many problems in chemistry, robotics or molecular biology can be expressed as a Distance CSP[4]. In this paper, we propose a specific methodology for tackling uncertainties in this class of problems. The main idea consists in solving the CSP without taking into account the uncertainties and using this information to approximate the solution space of the original CSP. This methodology permits to obtain an interior and exterior approximation of each solution sub-space by a set of disjoint boxes.

1 Introduction

Many of problems in chemistry, robotics or molecular biology can be expressed as a Distance CSP [1, 2]. Sometimes, the parameters of this kind of problems are determined in an experimental way, and therefore they have an uncertainty degree.

A classical approach for solving this class of problems is to solve the CSP without considering the uncertainties, and to obtain a set of solutions instead a set of solution sub-spaces, without knowing the real solution space. Another approach is to apply a branch and prune algorithm to generate a set of boxes that include all the solution sub-spaces. The disadvantage of this approach is that the algorithm generates a lot of boxes with a given precision and without information about independent solution sub-spaces. A subsequent task is to apply a clustering algorithm [3] to recover the solution sub-spaces.

In this paper, we propose a new methodology built from the combination of both approaches for tackling uncertainties in a Distance CSP. Two hypotheses are necessary to apply our algorithm:

- The problem has a finite number of solutions ρ, without taking into account the uncertainties.

[4] Constraint Satisfaction Problems involving distance constraints.

– The problem has only connected sets of solutions around each initial solution, when we consider the uncertainties.

The algorithm can be summarized in the following steps:

1. Finding all solutions of the problem without considering the uncertainties.
2. Applying a division of the initial space into sub-spaces containing only one initial solution each.
3. Applying a branch and prune algorithm, combined with a specialized feasibility checker to identify inner boxes, for each initial solution.

In the remainder of this section we explain the problem and its characteristics. Section 2 presents some basic definitions that we use in section 3 to explain our methodology. Section 4 shows some preliminary experiments and limitations of this approach. Finally, section 5 presents the conclusions of this work and future works.

1.1 Problem

A distance constraint c between two points P_i and P_j in a n-dimensional space $E \subset R^n$, can be expressed as follows:

$$c(P_i, P_j) : \sum_{k=1}^{n} (x_{ik} - x_{jk})^2 = d_{ij}^2 \tag{1}$$

where x_{ik} is the k-th coordinate of the point P_i, and d_{ij} is the distance value between them. A Distance CSP is a CSP formed by distance constraints. All fixed values in a Distance CSP, are called *parameters* of the problem. They can be fixed points or distances, and they can have real or interval values. When a CSP has parameters with interval values, it is called *CSP with uncertainties*.

We have identified two type of uncertainties: *soft uncertainty* and *hard uncertainty*. Soft uncertainty occurs when the independent parameter in the equation (the distance) has an interval value. Hard occurs when the parameters related with the variables have interval values (a fixed point, for example).

A main difficulty of the branch and prune algorithm is to determine when a box is totally included in the solution space. It is very important, since each branching in the search tree exponentially increases the number of boxes.

Xuan-Xu et al. [4, 5] propose techniques to verify inner boxes in CSP with non-isolated solutions. This type of techniques delivers good results in simple problems with soft uncertainty, but they are not effective in Distance CSP with hard uncertainty or several points in a space. In section 3.1 we propose a new algorithm to verify inner boxes in Distance CSP.

2 Background and Definitions

2.1 Interval Arithmetics

We use interval arithmetics to take into account the influence of rounding or uncertainties in computing calculation.

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

The set of intervals with bounds in $\mathbb{F}$, denoted by $\mathbb{I}$, is partially ordered by set inclusion. A Cartesian product of n intervals $B = I_1 \times \cdots \times I_n$ is called a *box*. Two boxes B_1 and B_2 are said disjoint if $B_1 \cap B_2 = \emptyset$.

Definition 2 (Set Extension) *Let S be a subset of $\mathbb{R}$. The approximation of S, denoted $\Box S$, is the smallest interval I such that $S \subseteq I$.*

An *interval extension* of $f : \mathbb{R}^n \to \mathbb{R}$ is a mapping $F : \mathbb{I}^n \to \mathbb{I}$ such that for all $I_1, \ldots, I_n \in \mathbb{I} : r_1 \in I_1, \ldots, r_n \in I_n \Rightarrow f(r_1, \ldots, r_n) \in F(I_1, \ldots, I_n)$. A more general description of interval extension and *natural interval extension* for a function and arithmetic operator can be found in [6, 7].

2.2 Constraint Programming

Definition 3 (CSP) *A CSP [8] is a triple (X, D, C) where $X = \{x_1, \ldots, x_n\}$ denotes a set of variables, $D = \{D_{x_1}, \ldots, D_{x_n}\}$ denotes a set of domains and $C = \{c_1, \ldots, c_m\}$ denotes a set of constraints.*

In this paper, we consider a special type of CSP called *Distance CSP with uncertainties*. It is a CSP with intervals domains (also called NCSP[5]), and therefore real-valued variables. Moreover, a *constraint* is an atomic formula built from a set of operations, a set of variables, a set $P = \{p_1, \ldots, p_k\}$ of parameters and a binary relation symbol of the set $\{=, \leq, \geq\}$. Uncertainties exist when one or more parameters have intervals values instead a real values. Let $Var(c)$ be the set of variables occurring in c, $Par(c)$ the set of parameters occurring in c, and ρ_c the relation associated to c, then $\rho_c = \{\boldsymbol{v} \in D_{x_1} \times \cdots \times D_{x_k}; x_1, \ldots x_k \in Var(c) \mid \exists \boldsymbol{p} \in p_1 \times \cdots \times p_j; p_1, \ldots, p_j \in Par(c), c(\boldsymbol{v}, \boldsymbol{p}) \text{ is verified } \}$. The global relation defined by the conjunction of all the constraints of a CSP, is denoted ρ.

2.3 Local Consistencies

Local consistencies, like *Hull-Consistency* or *Box-Consistency*, are extensively covered in literature (see [9], for example).

Definition 4 (Hull-Consistency) *A constraint c is said* hull consistent *with respect to a box B if and only if $B = \Box(\rho_c \cap B)$. A CSP $\mathbb{P}$ is hull consistent when all its constraints are hull consistent.*

Definition 5 (Inner Box) *Let c be a constraint and B a box. B is a inner box of c if and only if $B \subseteq \rho_c$.*

[5] Numeric Constraint Satisfaction Problem

3 Algorithm

We consider a Distance CSP with uncertainties $\mathbb{P}$. The first step in the methodology is solving the problem $\mathbb{P}'$ from $\mathbb{P}$ without considering uncertainties (by replacing each parameter with interval value by the middle point of the interval). We use a branch and prune algorithm (**solver**) for getting all solutions of $\mathbb{P}'$.

Let $S' = \{\boldsymbol{s_1}, \ldots, \boldsymbol{s_n}\}$ be the set of solutions for $\mathbb{P}'$. The next phase consists of applying a solution separation algorithm. For each solution $\boldsymbol{s_i} \in S'$, we calculate the equation of the median plane $Mp_{\boldsymbol{s_i},\boldsymbol{s_j}}(\boldsymbol{x})$ between $\boldsymbol{s_i}$ and $\boldsymbol{s_j}$, $\boldsymbol{s_i} \neq \boldsymbol{s_j}$. Algorithm 1 shows the Solution separation algorithm.

Algorithm 1 : SSA(Solution $\boldsymbol{s}$, Solutions S'): *constraints*

$C' \leftarrow \emptyset$
for all solution $\boldsymbol{s_i}$ in $S' \backslash \{\boldsymbol{s}\}$ **do**
  $Mp_{\boldsymbol{s},\boldsymbol{s_i}}(\boldsymbol{x}) \leftarrow (\boldsymbol{x} - \frac{\boldsymbol{s}+\boldsymbol{s_i}}{2}) \cdot (\frac{\boldsymbol{s}-\boldsymbol{s_i}}{\|\boldsymbol{s}-\boldsymbol{s_i}\|}) \geq 0$
  $C' \leftarrow C' \cup Mp_{\boldsymbol{s},\boldsymbol{s_i}}(\boldsymbol{x})$
return C'

For each solution $\boldsymbol{s_i} \in S'$, we solve a CSP $\mathbb{P}_{\boldsymbol{s_i}}$ built from the original CSP $\mathbb{P}$ and the set of equations $Mp_{\boldsymbol{s_i},\boldsymbol{s_j}}$, $\boldsymbol{s_i} \neq \boldsymbol{s_j}$. The sub-space for a solution $\boldsymbol{s_i}$, defined by the conjunction of all $Mp_{\boldsymbol{s_i},\boldsymbol{s_j}}$, $\boldsymbol{s_i} \neq \boldsymbol{s_j}$ is equivalent to the sub-space of a Voronoi diagram [10] for this point.

3.1 Feasibility checker

Our feasibility checker, used to determine inner boxes, is based on the following proposition:

Proposition 1 *Let $c : f(x_1, \ldots, x_k, p_1, \ldots, p_r) = D$ be a distance constraint with uncertainty ($D = [\underline{d}, \overline{d}]$, $p_1, \ldots, p_r \in P$ parameters). Let $B = I_{x_1} \times \cdots \times I_{x_k}$ be a box, and let $F_{\overline{p}}(B)$ be the interval extension of f when each variable x_i is replaced by the interval I_{x_i} and each interval-valued parameter p_i by the middle point of its value. If $F_{\overline{p}}(B) \subseteq D$ then B is an inner box of c.*

Proof. Considering $f_{\overline{p}}$, the function f when replacing each interval-valued parameter p_i by its middle point. By definition of interval extension, for any $x_1 \in I_{x_1}, \ldots, x_k \in I_{x_k}$ it follows that $f_{\overline{p}}(x_1, \ldots, x_k) \in F_{\overline{p}}(B)$, and if $F_{\overline{p}}(B) \subseteq D$ then there exists $d^* \in D$ such that $f_{\overline{p}} = d^*$, and then $B \subset \rho_c$.

Algorithm 2 : FCheck(CSP $\mathbb{P}$, Solution $\boldsymbol{s}$, Box B): *bool*

for all constraint $c(P_i, P_j)$ in $\mathbb{P}$ **do**
  if $F_{\overline{p}}(B) \not\subseteq D$ **then**
    return false
return true

We can see that if B is an inner box of c, for all c in $\mathbb{P}$ then $B \subseteq \rho$. The final branch and prune algorithm uses the feasibility checker (shown in algorithm 2) to identify an inner box each time when the pruning phase does not reduce the domains.

4 Preliminary Experiments

We have selected some toy problems to show the utility of this approach. The first one consists in determining 2 points in $\mathbb{R}^2$, given by the CSP $\mathbb{P}_1 : (X, D, C, P) = (\{\boldsymbol{P_1}, \boldsymbol{P_2}\}, \{\mathbb{R}^2, \mathbb{R}^2\}, \{d(\boldsymbol{P_1}, \boldsymbol{P_a}) = d_1^2, d(\boldsymbol{P_1}, \boldsymbol{P_b}) = d_2^2, d(\boldsymbol{P_2}, \boldsymbol{P_1}) = d_2^2, d(\boldsymbol{P_2}, \boldsymbol{P_b}) = d_2^2\}, \{P_a = (0,0), P_b = (6,0), d_1 = [4,5], d_2 = [3,4]\})$. The second one consists of determining a point in a Distance CSP with soft and hard uncertainties, $\mathbb{P}_2 : (X, D, C, P) = (\{\boldsymbol{P_1}\}, \{\mathbb{R}^2\}, \{d(\boldsymbol{P_1}, \boldsymbol{P_a}) = d_1^2, d(\boldsymbol{P_1}, \boldsymbol{P_b}) = d_2^2\}, \{P_a = (0,0), P_b = ([5.9, 6.1], [-0.1, 0.1]), d_1 = [4,5], d_2[3,4]\})$. Problem $\mathbb{P}_3$ is similar to $\mathbb{P}_2$ but with a parameter $P_a = ([-0.1, 0.1], [-0.2, 0.2])$. Table 1 shows the results[6] for these problems.

Prob.	**Sols.**	**Precision**	**Boxes**	**Inner**	**Time**	**FC Boxes**	**FC Inner**	**FC Time**
$\mathbb{P}_1$	4	0.2	32676	0	123.35s	29634	1594	91.60s
$\mathbb{P}_2$	2	0.01	47515	0	274.55s	13757	1073	16.95s
$\mathbb{P}_3$	2	0.01	62477	0	472.69s	30753	1027	106.98s

Table 1. Preliminary results of some toy problems

Column *Sols.* shows the number of solutions for the problem without considering the uncertainties. Column *Precision* shows the minimun interval width for applying a branching process over the domain of the variable. Columns *Boxes*, *Inner* and *Time* present the number of boxes, inner boxes and the computation time, respectively, for a branch and prune algorithm without feasibility checker. Columns *FC Box*, *FC inner*, *FC times* present the same values with feasibility checker. We can see that in all problems, the *FC* can help us for reducing the computation time and the number of boxes found.

4.1 Limitations

We have found some limitations of this methodology:

- Sometimes, it is not possible to calculate all solutions to a CSP $\mathbb{P}'$, because the problem without uncertainties has no solutions or because $\mathbb{P}'$ has a infinite number of solutions. In any of these cases, it is not possible to apply our algorithm.
- We consider that the number of solution sub-spaces in the CSP with uncertainties $\mathbb{P}$, is equal to the number of solutions of the CSP without uncertainty $\mathbb{P}'$. That is not necessarily true, since two solutions can share the same solution sub-space, or the problem $\mathbb{P}$ can have more solutions sub-spaces than

[6] Obtained on a Pentium III 500MHz 256MB RAM running IcosAlias v0.1c (http://www-sop.inria.fr/coprin/gchabert/icosalias.html) on a Linux kernel 2.4.27.

the solutions of $\mathbb{P}'$. The first case is easier to detect, because we can take two CSP $\mathbb{P}_{s_i}$ and $\mathbb{P}_{s_j}$, and seek for solutions on the median plane between s_i and s_j.

5 Conclusions

The contribution of this work is twofold: first, a methodology for solving a Distance CSP with uncertainties by approximating the solution sub-spaces without a posterior clustering algorithm; second, a new feasibility checker function for detecting inner boxes in a branch and prune algorithm. In spite of the limited experimentation, the preliminary results show the utility of this aproach. Although this methodology can be applied to Distance CSP with any type of uncertainty, the results of the feasibility checker are limited in the cases of hard uncertainty. At present we work on how to increase the performance of our FC function in the problems with hard uncertainty.

6 Acknowledgments

I would like to thank David Daney and Gilles Chabert for their valuable discusions and help during the execution of this work.

References

1. Krippahl, L., Barahona, P.: Applying constraint programming to protein structure determination. In: Principles and Practice of Constraint Programming. (1999)
2. Backofen, R.: Constraint techniques for solving the protein structure prediction problem. Lecture Notes in Computer Science **1520** (1998) 72–88
3. Vu, X.H., Sam-Haroud, D., Faltings, B.: Clustering for Disconnected Solution Sets of Numerical CSPs. In: Recent Advances in Constraints: Joint ERCIM/CoLogNET International Workshop on Constraint Solving and Constraint Logic Programming, CSCLP 2003, Budapest. Volume LNAI 3010., Springer-Verlag (2004)
4. Vu, X.H., Sam-Haroud, D., Silaghi, M.C.: Numerical constraint satisfaction problems with non-isolated solutions. In: 1st International Workshop on Global Constrained Optimization and Constraint Satisfaction (COCOS'2002), France, COCONUT Consortium (2002)
5. Vu, X.H., Sam-Haroud, D., Silaghi, M.C.: Approximation techniques for nonlinear problems with continuum of solutions. In: Proceedings of The 5th International Symposium on Abstraction, Reformulation and Approximation. Volume LNAI 2371., Canada (2002)
6. Moore, R.E.: Interval Analysis. Prentice Hall (1966)
7. Revol, N.: Introduction à l'arithmétique par intervalles. Rapport de Recherche 4297, INRIA Rhône-Alpes (2001)
8. Mackworth, A.K.: Consistency in networks of relations. Artificial Intelligence **8** (1977) 99–118
9. Benhamou, F., Goualard, F., Granvilliers, L., Puget, J.F.: Revising hull and box consistency. In: International Conference on Logic Programming. (1999) 230–244
10. Aurenhammer, F.: Voronoi diagrams-a survey of a fundamental geometric data structure (2001)