Propagation de contraintes sur les intervalles : application à l'estimation à erreurs bornées

Luc Jaulin

PROPAGATION DE CONTRAINTES SUR LES INTERVALLES APPLICATION A L'ESTIMATION A ERREURS BORNEES

L. Jaulin
Laboratoire d'Ingénierie des Systèmes Automatisés,
62 avenue Notre Dame du Lac,
F-49000 Angers, France.
jaulin@univ-angers.fr
http://www.istia.univ-angers.fr/~jaulin/

Mots clefs : Analyse par intervalles, calcul ensemblsite, erreurs bornées, estimation non-linéaire, propagation de contraintes.

Résumé : Le but de ce papier est de montrer que la propagation de contraintes combinée avec l'analyse par intervalles peut résoudre des problèmes d'équations et d'inéquations non linéaires comprenant de nombreuses inconnues en un temps très court. A titre d'exemple, nous considérerons le problème de l'estimation de paramètres pour un modèle paramétrique lorsque les valeurs des mesures non bruitées y_i et les temps non bruités associés t_i sont incertains mais compris entre deux valeurs connues.

1 Introduction

Un *problème de satisfaction de contraintes* (CSP pour *constraint satisfaction problem*) est constitué

1. d'un ensemble de variables réelles $x_1, \ldots, x_n$,
2. d'un ensemble de domaines (en général des intervalles) $[\check{x}_1], \ldots, [\check{x}_n]$ qui sont censés contenir les variables $x_1, \ldots, x_n$.
3. et un ensemble de contraintes reliant entre elles ces variables.

L'ensemble des solutions d'un CSP est défini comme l'ensemble des n-uplets $(x_1, \ldots, x_n)$ tels que

$$x_1 \in [\check{x}_1], \ldots, x_n \in [\check{x}_n]$$

et tels que toutes les contraintes soient satisfaites (voir par exemple (Hyvönen, 1992), (Granvilliers 1998) et (Jaulin 2000a) pour plus de détails).

Dans un problème d'estimation à erreurs bornées (voir (Walter E. and L. Pronzato, 1997) pour une introduction au sujet) , les variables x_i peuvent représenter n'importe quelle variable incertaine. Elles peuvent correspondre à une mesure non bruitée y_i (la mesure bruitée, étant connue sans aucune incertitude, ne peut être considérée comme variable), à un instant de mesure t_i (dans un contexte où les temps de mesure sont bruités), à une entrée u_i (dans un contexte à entrées bruitées), à une perturbation, ... Les domaines $[\check{x}_1], \ldots, [\check{x}_n]$ peuvent correspondre à des intervalles de mesures (barres erreurs) ou bien à une connaissance *a priori* de bornes sur certaines variables (perturbation non mesurée mais bornée). Les contraintes peuvent correspondre aux équations du modèle (par exemple $y_i = p_1 \exp(p_2 t_i)$), à des connaissances sur l'ordre du recueil des mesures $t_1 < t_2 < \cdots < t_m$, ou bien à des contraintes d'identifiabilité. Par exemple pour le modèle décrit par $y(t) = \exp(p_1 t) + \exp(p_2 t)$, on pourra poser $p_1 \geq p_2$ afin de le rendre identifiable.

Considérons, à titre d'exemple, le modèle décrit par les équations

$$y(t) = 20\exp(p_1 t) - 8\exp(p_2 t),$$

où t, $y(t), p_1$ et p_2 représentent respectivement le temps, la sortie à l'instant t et les paramètres du modèle. Supposons que 10 mesures y_1, ..., y_{10} ont été prélevées sur le système considéré au temps $t_1, \ldots, t_{10}$. Les y_i et les t_i sont supposés inconnus, mais bornés. Leurs domaines sont donnés par le tableau ci-dessous

i	$[\check{t}_i]$	$[\check{y}_i]$
1	$[0;1]$	$[7;12]$
2	$[1;7]$	$[0;2]$
3	$[1;7]$	$[6;7]$
4	$[3;4]$	$[-1;3]$
5	$[5;7]$	$[-9;-3]$
6	$[8;10]$	$[-5;-1]$
7	$[12;14]$	$[-4;3]$
8	$[16;18]$	$[-3;1]$
9	$[20;22]$	$[-3;1]$
10	$[24;26]$	$[-2;2]$

Ces domaines ont pu être obtenu à partir de mesures approximatives $\check{y}_i$ et les $\check{t}_i$ des y_i et les t_i et d'une connaissance de bornes sur les erreurs de mesure commises. Le modèle et les domaines choisis sont inspirés de (Jaulin and Walter 1999) ou une méthode par intervalles a été proposée pour estimer les paramètres d'un modèle dans un contexte où les temps de mesures ne sont connus de façon approximative. Les domaines pour les paramètres p_1 et p_2 sont donnés par

$$[\check{p}_1] = [-10000; 10000] \text{ et } [\check{p}_2] = [-10000; 10000],$$

ce qui revient à dire qu'aucune information n'est disponible *a priori* sur les paramètres p_1 et p_2. Les contraintes entre les variables sont données par les équations

$$\left\{ \begin{array}{lcl} y_1 & = & 20\exp(p_1 t_1) - 8\exp(p_2 t_1) \\ & \vdots & \\ y_{10} & = & 20\exp(p_1 t_{10}) - 8\exp(p_2 t_{10}) \end{array} \right.$$

qui nous viennent directement du modèle. La formulation du problème d'estimation peut donc assimilé à un CSP où les 22 variables sont les t_i, les y_i et les p_i.

Remarque 1 : On peut rajouter d'autres contraintes collatérales. Par exemple, si l'on sait que la mesure y_2 a été recueillies avant la mesure y_3 on rajoutera la contrainte $t_2 \leq t_3$. Le rajout de cette contrainte réduit l'ensemble des solutions. Nous montrerons dans la suite de ce papier que la prise en compte de cette contrainte rend l'ensemble des solutions du CSP vide. ■

La section 2 présente le principe de la propagation de contrainte lorsque les variables sont des nombres réels. La section 3 généralise l'approche au cas où les variables impliquées ont une nature vectorielle. Le but de ces deux sections est identique, à savoir la caractérisation efficace de l'ensemble des solutions d'un CSP.

2 Propagation de contraintes sur les intervalles

Les *méthodes de consistance* (appelée aussi méthodes par *propagation de contraintes*) basées sur les intervalles ont été initialement proposées par (Cleary, 1987) et (Davis, 1987). Elles permettent de réduire efficacement et simplement les domaines pour les variables d'un CSP sans jamais perdre de solutions. Parmi toutes les méthodes proposés, la plus simple consiste à décomposer l'ensemble des contraintes en contraintes primitives (par exemple $x_1 = x_2 x_3$, $x_1 = 3\sin(x_2)$, sont des contraintes primitives) et à contracter tant que cela est possible chacun des domaines relativement à chacune des contraintes primitives. Une analyse détaillée de l'approche peut être trouvée dans (Lhomme and M. Rueher, 1997) et (Granvilliers, 1998). Illustrons la méthode sur le problème exposé en introduction.

Chacune des 10 contraintes

$$y_i = 20\exp(p_1 t_i) - 8\exp(p_2 t_i)$$

peut de décomposer trois contraintes primitives :

$$\left\{ \begin{array}{lcl} a_i & = & 20\exp(p_1 t_i), \\ b_i & = & 8\exp(p_2 t_i), \\ y_i & = & a_i - b_i. \end{array} \right. \quad (1)$$

où les a_i et les b_i sont des variables auxiliaires. On dispose donc pour notre problème de $10 \times 3 = 30$ contraintes primitives. Chaque contrainte primitive peut être réécrite de plusieurs façons en isolant chacune des variables impliquées dans la contrainte. Par exemple, aux 3 contraintes primitives (1) on peut associer les 9 formes suivantes

$$\left\{ \begin{array}{lcl} a_i & = & 20\exp(p_1 t_i), \\ p_1 & = & \frac{1}{t_i}\ln\left(\frac{a_i}{20}\right) \\ t_i & = & \frac{1}{p_1}\ln\left(\frac{a_i}{20}\right) \\ b_i & = & 8\exp(p_2 t_i), \\ p_2 & = & \frac{1}{t_i}\ln\left(\frac{b_i}{8}\right) \\ t_i & = & \frac{1}{p_2}\ln\left(\frac{b_i}{8}\right) \\ y_i & = & a_i - b_i, \\ a_i & = & y_i + b_i \\ b_i & = & a_i - y_i \end{array} \right. \quad (2)$$

Ces équations sont appelées équations solutions. Nous avons donc 90 équations solutions pour notre problème. Maintenant, chacune de ces équations solutions peut être utilisée pour contracter les domaines pour les variables impliquées dans le CSP en utilisant

l'arithmétique des intervalles. Par exemple, si les domaines courants pour a_i, p_1 et t_i sont $[a_i], [p_1]$ et $[t_i]$, la première équation de (2) nous assure que si le domaine pour a_i est remplacé par

$$[a_i] = [a_i] \cap 20 \exp([p_1] * [t_i])$$

aucune solution n'est perdue. En appliquant ce type de contraction, à tour de rôle pour chacune des 90 équations solutions, on finit par atteindre un équilibre. Souvent, cette est très rapide, mais les domaines obtenus pour les variables ne sont pas toujours les plus petits possibles, c'est-à-dire qu'on arrive une une situation de blocage.

Reprenons l'exemple présenté en introduction et appliquons la propagation de contraintes exposée ci-dessus. En 0.004 secondes, sur un Pentium 300, les domaines obtenus pour les paramètres sont

$$[\hat{p}_1] = [-0.723; 0.648] \text{ et } [\hat{p}_2] = [-0.767; 0.182],$$

qui sont sensiblement plus petits que les domaines initiaux. Malheureusement, les domaines pour les t_i et les y_i n'ont pas été contractés. Si maintenant nous rajoutons la contrainte $t_2 \leq t_3$ (voir remarque 1) nous obtenons en 0.001 seconde que l'ensemble des solutions est vide. Cela signifie que la contrainte $t_2 \leq t_3$ est incompatible avec nos domaines *a priori* pour les y_i et les t_i.

Revenons à la situation où la contrainte $t_2 \leq t_3$ n'est pas prise en compte. Pour se sortir de la situation de blocage, autorisons-nous à couper certains domaines en plusieurs sous intervalles. Effectuons alors les contractions par propagation sur toutes les combinaisons possibles de ces sous-domaines et réunissons tous les domaines ainsi contractés. Nous pouvons espérer ainsi obtenir des domaines plus précis pour les variables.

En effet, si seulement les domaines pour p_1 et p_2 sont découpés, après 10 coupures représentées sur la figure 1, nous obtenons en 0.06 secondes les domaines

$$[\hat{p}_1] = [0,290, 0,422] \text{ et } [\hat{p}_2] = [0,052, 0,107]$$

pour p_1 et p_2. Aucune amélioration n'est obtenue si p_1 et p_2 sont coupés en sous-intervalles encore plus petit.

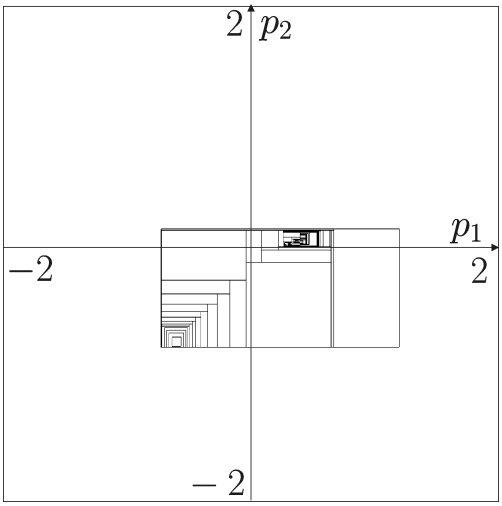


FIG 1 - Bissections et contractions engendrées dans l'espace (p_1, p_2)

Les domaines pour les t_i et les y_i sont représentés sur la figure 2.

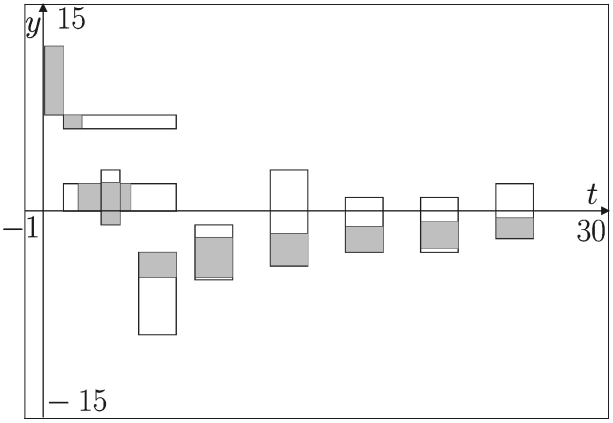


FIG 2 - Domaines initiaux et domaines contractés pour les temps de mesures t_i et pour les sorties mesurées y_i (t_i et y_i sont incertains) ;

Les méthodes de propagation de contraintes sur les intervalles, permettent n'obtenir rapidement des intervalles relativement petits et contenant à coup sur tout les valeurs consistantes avec les intervalles initiaux et avec les contraintes. Cependant, elle ne permettent pas d'évaluer si ces intervalles sont les plus petits possibles. Une solution envisageable est de valider les contractions par une recherche intérieure (Jaulin 2000b). Une autre solution plus coûteuse mais plus générale consiste à généraliser la propagation de contraintes sur les intervalles en propagation de contraintes sur sous-ensembles (Jaulin, Kieffer, Braems,

Walter, 2001) de $\mathbb{R}^n$. Cette approche est présentée dans le prochain paragraphe.

3 Propagation de contraintes sur les sous-pavages

Les intervalles (ou les union d'intervalles) permettent d'approximer les domaines pour les variables de $\mathbb{R}$, comme celles impliquées dans les CSPs. Lorsque les variables manipulées sont vectorielles, les domaines pour ces variables sont des sous-ensembles de $\mathbb{R}^n$ qui peuvent être approximés par des sous-pavages (voir (Jaulin, 1994) et (Kieffer, 1999)). Les sous-pavages de $\mathbb{R}^n$ sont des unions de pavés de $\mathbb{R}^n$. Il existe des algorithmes efficaces pour calculer leur image directe ou inverse par une fonction non-linéaire $\mathbf{f}$. Par exemple, un sous pavage associé à l'ensemble

$$\mathbb{X} \triangleq \left\{(x_1, x_2) \mid 1 \leq x_1^2 + x_2^2 + x_1 x_2 \leq 2\right\}$$

est donné par la figure 3. Les pavés gris foncés sont à l'intérieur de $\mathbb{X}$ et les pavés gris-clairs sont à l'extérieur de $\mathbb{X}$. On ne sais rien concernant les pavés blancs.

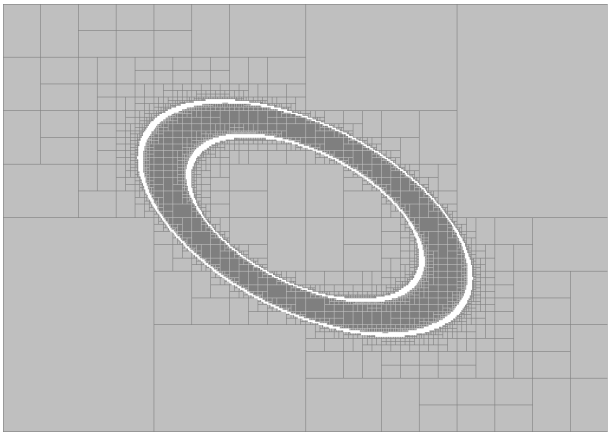


FIG 3 - Sous pavage associé à l'ensemble $\mathbb{X} \triangleq \left\{(x_1, x_2) \mid 1 \leq x_1^2 + x_2^2 + x_1 x_2 \leq 2\right\}$.

Si nous considérons maintenant la fonction de $\mathbb{R}^2$ vers $\mathbb{R}^2$ donnée par

$$\mathbf{f} : \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \rightarrow \begin{pmatrix} \sin(x_1) + x_2^2 \\ x_1^2 + \sin(x_2) \end{pmatrix}$$

L'ensemble $\mathbf{f}^{-1}(\mathbb{X})$ peut être représenté par le sous-pavage de la figure 4.

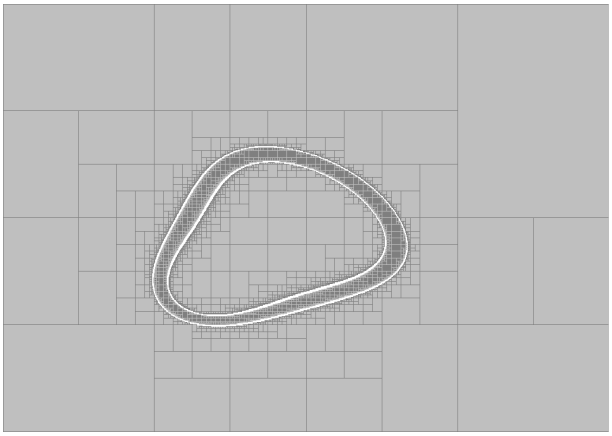


FIG 4 - Sous pavage encadrant l'ensemble $\mathbf{f}^{-1}(\mathbb{X})$

Ce sous-pavage a été obtenu par l'algorithme d'inversion ensembliste SIVIA (pour Set Inversion Via Interval Analysis (Jaulin, 1994)) en 0.5 secondes.

En utilisant les sous pavages, il est possible d'étendre les techniques associées aux CSPs à variables réelles aux CSPs à variables vectorielles (Jaulin, Kieffer, Braems, Walter, 2001).

Un *problème de satisfaction de contraintes* sera alors constitué

1. d'un ensemble de variables vectorielles $\mathbf{x}_1, \ldots, \mathbf{x}_n$,
2. d'un ensemble de sous ensembles de $\mathbb{R}^n$ (représentés par des sous pavages) $\mathbb{X}_1, \ldots, \mathbb{X}_n$ qui sont censés contenir les variables $\mathbf{x}_1, \ldots, \mathbf{x}_n$.
3. et un ensemble de contraintes reliant entre-elles ces variables (en général, ces contraintes sont de la forme $\mathbf{x}_i = \mathbf{f}(\mathbf{x}_j)$).

L'ensemble des solutions de ce CSP sera donc l'ensemble des n-uplets $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ tels que $\mathbf{x}_1 \in \mathbb{X}_1, \ldots, \mathbf{x}_n \in \mathbb{X}_n$ et tels que toutes les contraintes soient satisfaites. Un vecteur $\mathbf{x}_i$ est dit consistant avec le CSP, si on peut instancier les autres vecteurs dans leur domaine respectif, de tels façon que toutes les contraintes soient satisfaites. Notons $\hat{\mathbb{X}}_i$ l'ensemble des $\mathbf{x}_i$ consistants avec le CSP. Dans certain cas, on sait calculer les ensembles $\hat{\mathbb{X}}_i$.

Considérons par exemple le système de contraintes

$$\begin{cases} \mathbf{x}_1 &= \mathbf{f}_3(\mathbf{x}_3), \\ \mathbf{x}_4 &= \mathbf{f}_4(\mathbf{x}_2), \\ \mathbf{x}_2 &= \mathbf{f}_5(\mathbf{x}_5), \\ \mathbf{x}_2 &= \mathbf{f}_2(\mathbf{x}_1). \end{cases}$$

Associons à ce système de contraintes le graphe dessiné sur la figure 5.

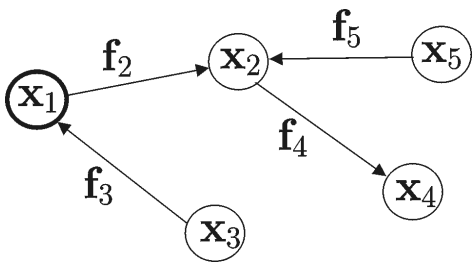


FIG 5 - Graphe associé au CSP ;
ce graphe est un arbre
la racine choisie pour l'arbre est $\mathbf{x}_1$

Dans ce cas particulier ce graphe est arbre car il ne contient aucun cycle. Or, pour un arbre, on sait obtenir les plus petit domaines pour $\mathbf{x}_1, \ldots, \mathbf{x}_5$ en utilisant un algorithme de propagation-rétropropagation. Pour cela, il suffit de choisir un noeud, par exemple $\mathbf{x}_1$ et de contracter les domaines de chacune des variables a partir des feuilles jusqu'à la racine puis de la racine jusqu'au feuilles. Dans notre exemple, on obtient l'algorithme ensembliste de la Table 1.

entrées :	$\mathbb{X}_1, \mathbb{X}_2, \mathbb{X}_3, \mathbb{X}_4, \mathbb{X}_5$;
1	$\mathbb{X}_2 := \mathbb{X}_2 \cap \mathbf{f}_5(\mathbb{X}_5)$;
2	$\mathbb{X}_2 := \mathbb{X}_2 \cap \mathbf{f}_4^{-1}(\mathbb{X}_4)$;
3	$\mathbb{X}_1 := \mathbb{X}_1 \cap \mathbf{f}_3(\mathbb{X}_3)$;
4	$\mathbb{X}_1 := \mathbb{X}_2 \cap \mathbf{f}_2^{-1}(\mathbb{X}_2)$;
5	$\mathbb{X}_2 := \mathbb{X}_2 \cap \mathbf{f}_2(\mathbb{X}_1)$;
6	$\mathbb{X}_3 := \mathbb{X}_3 \cap \mathbf{f}_3^{-1}(\mathbb{X}_1)$;
7	$\mathbb{X}_5 := \mathbb{X}_5 \cap \mathbf{f}_5^{-1}(\mathbb{X}_2)$;
8	$\mathbb{X}_4 := \mathbb{X}_4 \cap \mathbf{f}_4(\mathbb{X}_2)$;
sorties :	$\mathbb{X}_1, \mathbb{X}_2, \mathbb{X}_3, \mathbb{X}_4, \mathbb{X}_5$.

Table 1 - Principe de l'algorithme de propagation-rétropropagation

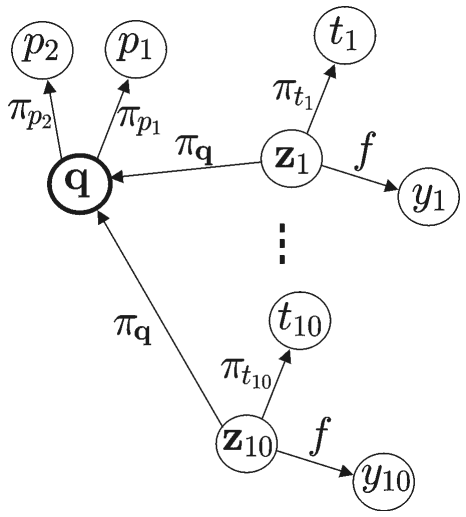
Cet algorithme peut être approximer de façon très précise grâce aux sous-pavages.

Si le graphe des contraintes n'est pas un arbre (c'est le cas si les contraintes ne sont pas binaires ou s'il existe des cycles dans le graphe), on peut toujours regrouper les variables en un seul vecteur afin que le graphe devienne un arbre. C'est le principe des techniques de *clustering* (ou de *rassemblement*) présentées dans (Dechter and Pear, 1989) dans un contexte où les domaines pour les variables sont des ensembles discrets.

Appliquons cette technique de rassemblement au problème d'estimation présenté en introduction. On fabriquant, les vecteurs $\mathbf{q} = (p_1, p_2)$ et $\mathbf{z}_i = (\mathbf{q}, t_i)$, les contraintes reliant les variables de notre problème peuvent s'écrire sous la forme:

$$\begin{cases} p_1 = \pi_{p_1}(\mathbf{q}) \\ p_2 = \pi_{p_2}(\mathbf{q}) \\ \mathbf{q} = \pi_{\mathbf{q}}(\mathbf{z}_1) \\ t_1 = \pi_{t_1}(\mathbf{z}_1) \\ y_1 = f(\mathbf{z}_1) \\ \quad \vdots \\ \mathbf{q} = \pi_{\mathbf{q}}(\mathbf{z}_{10}) \\ t_{10} = \pi_{t_{10}}(\mathbf{z}_{10}) \\ y_{10} = f(\mathbf{z}_{10}) \end{cases} \tag{3}$$

π_v est l'opérateur de projection sur la variable v et où $f(p_1, p_2, t) = 20\exp(p_1 t) - 8\exp(p_2 t)$. L'arbre des contraintes se trouve dessiné sur la figure 6.



Arbre associé aux contraintes du problème d'estimation après rassemblement des variables

Dans ce contexte, en prenant comme noeud la variable $\mathbf{q}$, l'algorithme de propagation-rétropropagation s'écrit comme indiqué sur la Table 2. Dans cet algorithme les entrées et les sorties sont des sous-ensembles de $\mathbb{R}$ (par exemple des intervalles). $\mathbb{Q}$ est un sous-ensemble de $\mathbb{R}^2$ et représente le domaine pour $\mathbf{q}$. Les $\mathbb{Z}_i$ sont des sous-ensembles de $\mathbb{R}^3$ représentent les domaines pour les $\mathbf{z}_i$.

entrées :	$[p_1], [p_2], [t_1], \dots [t_{10}], [y_1], \dots [y_{10}]$
1	$\mathbb{Q} := \pi_{p_1}^{-1}([p_1])$;
2	$\mathbb{Q} := \mathbb{Q} \cap \pi_{p_2}^{-1}([p_2])$;
3	for $i := 1$ to 10
4	$\mathbb{Z}_i := \pi_{t_i}^{-1}([t_i])$;
5	$\mathbb{Z}_i := \mathbb{Z}_i \cap f^{-1}([y_i])$;
6	$\mathbb{Q} := \mathbb{Q} \cap \pi_q(\mathbb{Z}_i)$;
7	end for
8	for $i := 1$ to 10
9	$\mathbb{Z}_i := \mathbb{Z}_i \cap \pi_q^{-1}(\mathbb{Q})$;
10	$[y_i] := [y_i] \cap f(\mathbb{Z}_i)$;
11	$[t_i] := [t_i] \cap \pi_{t_i}(\mathbb{Z}_i)$;
12	end for
13	$[p_1] := [p_1] \cap \pi_{p_1}(\mathbb{Q})$;
14	$[p_2] := [p_2] \cap \pi_{p_2}(\mathbb{Q})$;
sorties :	$[p_1], [p_2], [t_1], \dots [t_{10}], [y_1], \dots [y_{10}]$

Table 2 - Algorithme de propagation-rétropropagation pour le problème d'estimation

Si nous programmions cet algorithme avec le calcul sur les sous-pavages, nous obtiendrions, pour une approximation convenable, un temps de calcul de l'ordre de quelques minutes. Ce temps de calcul est très grand comparé aux méthodes exposées en section 2 sur l'exemple considéré. Cependant, pour des problèmes plus complexes avec de nombreuses variables, une méthode de propagation de contraintes sur les intervalles arrive souvent à une approximation trop pessimiste, et il ne nous reste plus qu'à découper suivant toutes les directions. Le temps de calcul devient alors beaucoup trop grand pour rendre l'approche inutilisable. En revanche, dans des conditions très larges, une méthode de regroupement se traduit par la manipulation d'ensembles de dimensions inférieures ou égale à trois (si les regroupements ont été faits convenablement) (voir (Sam-Haroud, 1995) et (Lottaz, 2000)). Ceci permet d'obtenir une caractérisation convenable pour les solutions du CSP en un temps très grand mais encore acceptable et pour une classe très grande classe de CSP.

La grande classe des CSPs, dont nous venons de parler, comprend les CSPs décomposables en contraintes binaires ou ternaires qui sont convexes par rapport aux intervalles. C'est le cas par exemple des CSPs formés de contraintes du type $f(\mathbf{x}) = 0$ où f ne comprend que les additions, des multiplications et des fonctions croissantes (exp, log, ...). L'idée sous-jacente est qu'une contrainte binaire (par exemple $x_2 = \sin(x_1)$) et une contrainte ternaire (par exemple $x_5 \geq x_3.x_4$) peuvent être respectivement approximées arbitrairement par des sous pavages de dimension deux et trois.

4 Conclusion

Ce papier montre comment l'analyse par intervalles, combinée aux techniques de propagation de contraintes, peut résoudre efficacement des problèmes non-linéaires avec de nombreuses inconnues. Une illustration de l'approche utilisée a été donnée dans le contexte de l'estimation à erreurs bornées ou les inconnues sont à la fois les valeurs des paramètres inconnus, les temps de mesures incertains et les valeurs non-bruités des mesures, elles-aussi incertaines ('incertain' signifiant ici : 'inconnu mais borné'). De plus, ce papier montre les liens étroits qu'il existe entre l'estimation à erreurs bornées et les problèmes de satisfaction de contraintes (ou CSP) très étudiés dans les milieux de l'informatique et de l'intelligence artificielle. De nombreux résultats existants pour les CSPs ne demandent qu'à être appliqués au contexte de l'estimation à erreurs bornées.

Les programmes d'estimation associés à l'exemple présenté dans ce papier ont été programmés en Borland C++ builder III. Si vous me les demandez, je serais heureux de vous donner les sources de ces programmes avec les quelques explications qui vont avec.

Références

Cleary, J.C. (1987). Logical arithmetic, *Future Computing Systems*, **2**(2), 125-149.

Davis, E. (1987). Constraint propagation with interval labels, *Artificial Intelligence*, **32**, 281-331.

Dechter, R. and J. Pearl (1989). Tree-clustering for constraint networks, *Artificial Intelligence*, **38**(3), 353.–366.

Jaulin, L. (1994). Solution globale et garantie de problèmes ensemblistes, application à l'estimation non linéaire et à la commande robuste, thèse de l'université de Paris-Sud, disponible à l'adresse *http://istia/~jaulin/thesejaulin.zip*.

Jaulin, L. (2000a). Le calcul ensembliste par analyse

par intervalles, Habilitation à diriger des recherches de l'université de Paris-Sud, disponible à l'adresse *http://istia/~jaulin/hdrjaulin.zip*.

Jaulin, L. (2000b). Interval constraint propagation with application to bounded-error estimation, *Automatica*, **36**, 1547-1552.

Jaulin, L. and E. Walter (1999). Guaranteed bounded-error parameter estimation for nonlinear models with uncertain experimental factors, *Automatica*, **35**, 849-856

Jaulin, L., M. Kieffer, I. Braems and E. Walter (2001). Guaranteed nonlinear estimation using constraint propagation on sets, accepté à International Journal of Control.

Granvilliers, L. (1998). Consistances locales et transformations symboliques de contraintes d'intervalles, thèse de l'université d'Orléans.

Hyvönen, E. (1992). Constraint reasoning based on interval arithmetic; the tolerance propagation approach, *Artificial Intelligence*, **58**, 71-112.

Kieffer, M. (1999). Estimation ensembliste par analyse par intervalles, application à la localisation d'un véhicule, thèse de l'université de Paris-Sud.

Lhomme, O. and M. Rueher (1997). Application des techniques CSP au raisonnement sur les intervalles, *Revue d'intelligence artificielle*, **11**(3), 283-311.

Lottaz, C. (2000). Collaborative design using solution spaces, thèse No. 2119 de EPFL.

Sam-Haroud, D. (1995). Constraint consistency techniques for continuous domains, thèse No. 1423 EPFL.

Walter, E. and L. Pronzato (1997). Identification of parametric models from experimental data. *Springer-Verlag*, Londre.