# Propagation de contraintes sur les intervalles pour l'estimation ensembliste

**Luc Jaulin**

*Laboratoire d'Ingénierie des Systèmes Automatisés,*
*62 avenue Notre Dame du Lac,*
*F-49000 Angers*
*jaulin@univ-angers.fr*

RÉSUMÉ. *Le but de cet article est de montrer que la propagation de contraintes combinée avec l'analyse par intervalles peut traiter 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 sorties $y_i$ que l'on a mesurées et les temps de mesure associés $t_i$ sont incertains mais compris entre deux valeurs connues.*

ABSTRACT. *This paper shows that constraint propagation combined with interval analysis can efficiently characterize the solution set of difficult problems of nonlinear equalities and inequalities even when the number of variables is large. As an example, we shall consider the problem of parameter estimation of a parametric model when the values of the output-data and times at which they were collected are uncertain.*

MOTS-CLÉS : *analyse par intervalles, calcul ensembliste, erreurs bornées, estimation non linéaire, propagation de contraintes.*

KEYWORDS: *bounded errors, constraint propagation, nonlinear estimation, interval analysis, set computation.*

APII-JESA – 36/2002. Identification des systèmes, pages 383 à 395

## 1. Introduction

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

1) d'un ensemble de variables scalaires $x_1, \ldots, x_n$,

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) d'un ensemble de contraintes reliant entre elles ces variables.

Comme nous allons l'illustrer dans cet article, cette notion, classique en informatique (voir par exemple [GRA 98, HYV 92, JAU 00b] pour plus de détails), permet une représentation unifiée (voir [JAU 01b], chapitre 6) d'une grande classe de problèmes d'estimation à erreurs bornées. 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.

Dans un problème d'estimation à erreurs bornées (voir [WAL 97] pour une introduction au sujet), les variables $x_i$ peuvent représenter n'importe quelle variable incertaine. Elles peuvent correspondre à la valeur d'une sortie $y_i$ mesurée à un instant de mesure $t_i$ (dans un contexte où les temps de mesure sont incertains), à une entrée $u_i$ (dans un contexte à entrées bruitées), à une perturbation, etc. 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). Notons qu'une variable parfaitement connue aura pour domaine un singleton. On pourra, si on le souhaite, la retirer de l'ensemble des variables et elle prendra alors le statut de constante. 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 dix mesures des variables $y_1, \ldots, y_{10}$ aient 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 *a priori* sont donnés par le tableau 1 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 | $[24; 26]$ | $[-2; 2]$ |

**Tableau 1**. *D*omaines a priori pour le problème d'estimation

Ces domaines ont pu être obtenus à partir de mesures approximatives $\check{y}_i$ et $\check{t}_i$ des quantités $y_i$ et $t_i$ et d'une connaissance des bornes sur les erreurs de mesure commises. Le modèle et les domaines choisis sont inspirés de [JAU 99] où 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 que 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'une grande incertitude existe *a priori* sur les paramètres $p_1$ et $p_2$. Les contraintes entre les variables sont données par les équations :

$$\begin{cases} y_1 &= 20\exp(p_1 t_1) - 8\exp(p_2 t_1) \\ &\vdots \\ y_{10} &= 20\exp(p_1 t_{10}) - 8\exp(p_2 t_{10}) \end{cases}$$

qui résultent directement du modèle. Le problème d'estimation peut donc être assimilé à un CSP où les 22 variables sont les $t_i$, les $y_i$ et les $p_i$.

REMARQUE. — On peut rajouter des contraintes collatérales. Par exemple, si l'on sait que la mesure $y_2$ a été recueillie avant la mesure $y_3$, on rajoutera la contrainte $t_2 \leq t_3$, ce qui réduit l'ensemble des solutions. Nous observerons que la prise en compte de cette contrainte rend l'ensemble des solutions du CSP de notre exemple vide.

La section 2 présente le principe de la propagation de contraintes 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* (appelées aussi très souvent méthodes par *propagation de contraintes*) exploitant le calcul par intervalles ont été initialement proposées par Cleary [CLE 87] et Davis [DAV 87]. Elles permettent de contracter 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. Rappelons qu'une contrainte primitive est une contrainte binaire ou bien ternaire, pour laquelle chacune des deux ou trois variables n'apparaît qu'une seule fois dans l'expression de la contrainte. Par exemple $x_1 = 3 + x_2 x_3$, $x_1 = 3\sin(x_2)$, sont deux contraintes primitives. Une fois cette décomposition faite, on contracte tant que cela est possible chacun des domaines relativement à chacune des contraintes primitives. Une analyse détaillée de l'approche peut être trouvée dans [GRA 98, LHO 97]. Illustrons la méthode sur le problème exposé en introduction.

Chacune des dix contraintes :

$$y_i = 20\exp\left(-p_1 t_i\right) - 8\exp\left(-p_2 t_i\right) \tag{1}$$

peut se décomposer trois contraintes primitives :

$$\left\{\begin{array}{lcl} a_i & = & 20\exp\left(-p_1 t_i\right) \\ b_i & = & 8\exp\left(-p_2 t_i\right) \\ y_i & = & a_i - b_i \end{array}\right. \tag{2}$$

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. Chacune peut être réécrite de plusieurs façons en isolant chacune des variables impliquées. Par exemple, aux trois contraintes primitives (2), on peut associer les neuf formes suivantes :

$$\left\{\begin{array}{lcl} a_i & = & 20\exp\left(-p_1 t_i\right) \\ p_1 & = & \frac{-1}{t_i}\ln\left(\frac{a_i}{20}\right) \\ t_i & = & \frac{-1}{p_1}\ln\left(\frac{a_i}{20}\right) \\ b_i & = & 8\exp\left(-p_2 t_i\right) \\ p_2 & = & \frac{-1}{t_i}\ln\left(\frac{b_i}{8}\right) \\ t_i & = & 0\frac{-1}{p_2}\ln\left(\frac{b_i}{8}\right) \\ y_i & = & a_i - b_i \\ a_i & = & y_i + b_i \\ b_i & = & a_i - y_i \end{array}\right. \tag{3}$$

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 (3) 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 procédure est très rapide, mais les domaines obtenus pour les variables ne sont pas toujours les plus petits possibles. On arrive ainsi à une situation de blocage.

Reprenons l'exemple présenté en introduction et appliquons la propagation de contraintes exposée ci-dessus. En 0,004 seconde, 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 la remarque à la fin de la section 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 donnés *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 découper certains domaines en plusieurs sous-intervalles. Nous pouvons espérer ainsi obtenir des domaines plus précis pour les variables. La stratégie que nous allons utiliser (appelée stratégie *max-dom* dans le contexte des CSP) consiste à couper les pavés de plus grande longueur, perpendiculairement à leur côté de plus long. Si seulement les domaines pour $p_1$ et $p_2$ sont découpés, après dix coupures représentées sur la figure 1, nous obtenons en 0,06 seconde 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 petits. Les deux bissections, visibles à cette échelle, sont représentées en pointillés.

Les domaines contractés pour les $t_i$ et les $y_i$ sont représentés sur la figure 2. Sur cette figure, les pavés blancs représentent les domaines initiaux pour les couples $(t_i, y_i)$ alors que les pavés gris représentent les domaines contractés.

Les méthodes de propagation de contraintes sur les intervalles permettent d'obtenir rapidement des intervalles relativement petits et contenant à coup sûr toutes les valeurs consistantes avec les intervalles initiaux et avec les contraintes. Cependant, elles ne permettent pas d'évaluer la distance entre les intervalles obtenus et ceux que l'on cherche (c'est-à-dire les plus petits intervalles qui contiennent toutes les valeurs consistantes). Une solution envisageable est d'effectuer une recherche intérieure [JAU 00a] et d'utiliser les résultats de cette recherche comme critère d'arrêt pour les

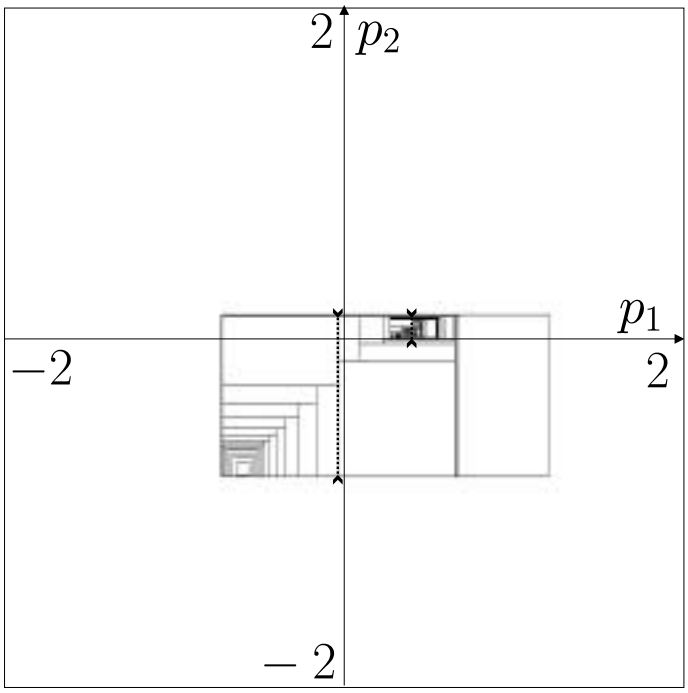


**Figure 1.** *Bissections et contractions engendrées dans l'espace* $(p_1, p_2)$

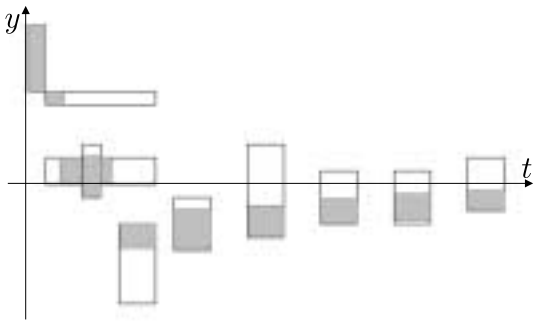


**Figure 2.** *Domaines initiaux et domaines contractés pour les temps de mesures* $t_i$ *et pour les sorties mesurées* $y_i$ *(*$t_i$ *et* $y_i$ *sont incertains)*

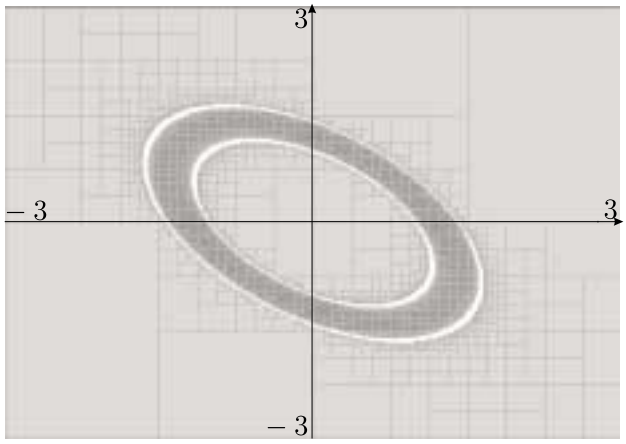


**Figure 3.** *Sous-pavages utilisés pour encadrer le sous-ensemble* $\mathbb{X} \triangleq \left\{(x_1, x_2) \,|1 \leq x_1^2 + x_2^2 + x_1 x_2 \leq 2\right\}$

méthodes de propagation. Une autre solution plus coûteuse mais plus générale consiste à étendre aux sous-ensembles de $\mathbb{R}^n$ la propagation de contraintes sur les intervalles [JAU 01a]. Cette approche est présentée dans la prochaine section.

## 3. Propagation de contraintes sur les sous-pavages

Les intervalles (ou les unions d'intervalles) permettent d'approximer les domaines pour les variables de $\mathbb{R}$, comme celles impliquées dans les CSP. 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 [JAU 94, KIE 99]). Les sous-pavages de $\mathbb{R}^n$ sont des unions de pavés disjoints 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 :

$$\mathbb{X} \triangleq \left\{(x_1, x_2) \in \mathbb{R}^2 \mid 1 \leq x_1^2 + x_2^2 + x_1 x_2 \leq 2\right\} \tag{4}$$

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

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 les sous-pavages de la figure 4.

Ces sous-pavages ont été obtenus par l'algorithme d'inversion ensembliste SIVIA (pour *Set Inversion Via Interval Analysis* [JAU 94]) en 0,5 seconde.

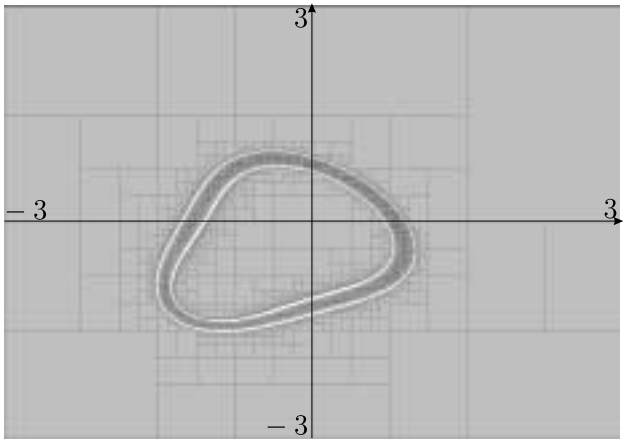


**Figure 4.** *Sous-pavages encadrant l'ensemble* $\mathbf{f}^{-1}(\mathbb{X})$

En utilisant les sous-pavages, il est possible d'étendre les techniques associées aux CSP à variables scalaires aux CSP à variables vectorielles [JAU 01a]. 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) d'un ensemble de contraintes reliant 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 l'on peut instancier les autres vecteurs dans leur domaine respectif, de telle façon que toutes les contraintes soient satisfaites. Notons $\hat{\mathbb{X}}_i$ l'ensemble des $\mathbf{x}_i$ consistants avec le CSP. Dans certains cas, on sait calculer les ensembles $\hat{\mathbb{X}}_i$.

Considérons par exemple le système de contraintes :

$$\left\{ \begin{array}{lcl} \mathbf{x}_1 & = & \mathbf{f}_3(\mathbf{x}_3) \\ \mathbf{x}_4 & = & \mathbf{f}_4(\mathbf{x}_2) \\ \mathbf{x}_2 & = & \mathbf{f}_5(\mathbf{x}_5) \\ \mathbf{x}_2 & = & \mathbf{f}_2(\mathbf{x}_1) \end{array} \right. \tag{5}$$

Ce système de contraintes correspond au graphe de la figure 5.

Dans ce cas particulier, ce graphe est un arbre car il ne contient aucun cycle. Or, pour un arbre, on sait obtenir les plus petits domaines pour $\mathbf{x}_1, \ldots, \mathbf{x}_5$ en utilisant un algorithme de propagation-rétropropagation (voir par exemple [JAU 01b]). Pour cela,

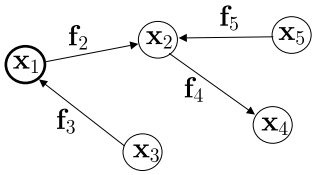


**Figure 5.** *Graphe associé au CSP ; ce graphe est un arbre ; la racine choisie pour l'arbre est* $\mathbf{x}_1$

il suffit de choisir un nœud comme la racine de l'arbre, par exemple $\mathbf{x}_1$, et de contracter les domaines de chacune des variables à partir des feuilles jusqu'à la racine puis de la racine jusqu'aux feuilles. Dans notre exemple, on obtient l'algorithme ensembliste du tableau 2.

| | |
|---|---|
| entrées : | $\mathbb{X}_1, \mathbb{X}_2, \mathbb{X}_3, \mathbb{X}_4, \mathbb{X}_5$ ; |
| 1 | $\mathbb{X}_2 := \mathbb{X}_2 \cap \mathbf{f}_5\left(\mathbb{X}_5\right)$ ; // début de la propagation |
| 2 | $\mathbb{X}_2 := \mathbb{X}_2 \cap \mathbf{f}_4^{-1}\left(\mathbb{X}_4\right)$ ; |
| 3 | $\mathbb{X}_1 := \mathbb{X}_1 \cap \mathbf{f}_3\left(\mathbb{X}_3\right)$ ; |
| 4 | $\mathbb{X}_1 := \mathbb{X}_1 \cap \mathbf{f}_2^{-1}\left(\mathbb{X}_2\right)$ ; // fin de la propagation |
| 5 | $\mathbb{X}_2 := \mathbb{X}_2 \cap \mathbf{f}_2\left(\mathbb{X}_1\right)$ ; // début de la rétropropagation |
| 6 | $\mathbb{X}_3 := \mathbb{X}_3 \cap \mathbf{f}_3^{-1}\left(\mathbb{X}_1\right)$ ; |
| 7 | $\mathbb{X}_5 := \mathbb{X}_5 \cap \mathbf{f}_5^{-1}\left(\mathbb{X}_2\right)$ ; |
| 8 | $\mathbb{X}_4 := \mathbb{X}_4 \cap \mathbf{f}_4\left(\mathbb{X}_2\right)$ ; // fin de la rétropropagation |
| sorties : | $\mathbb{X}_1, \mathbb{X}_2, \mathbb{X}_3, \mathbb{X}_4, \mathbb{X}_5$. |

**Tableau 2**. *P*rincipe de l'algorithme de propagation-rétropropagation

Cet algorithme peut être approximé 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 certaines variables afin que le graphe devienne un arbre. C'est le principe des techniques de *clustering* (ou de rassemblement) présentées dans [DEC 89] dans un contexte où les domaines pour les variables sont des ensembles discrets.

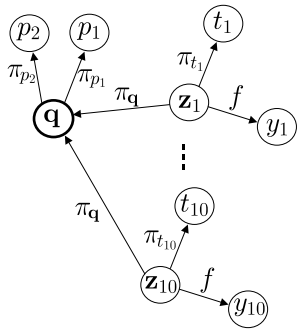


**Figure 6.** *Arbre associé aux contraintes du problème d'estimation après rassemblement des variables*

Appliquons cette technique de rassemblement au problème d'estimation présenté en introduction. En 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 :

$$\left\{\begin{array}{l} p_1 = \pi_{p_1}(\mathbf{q}) \\ p_2 = \pi_{p_2}(\mathbf{q}) \\ \mathbf{q} = \pi_{\mathbf{q}}(\mathbf{z}_1) \\ t_1 = \pi_{t_1}(\mathbf{z}_1) \\ y_1 = f(\mathbf{z}_1) \\ \quad \vdots \\ \mathbf{q} = \pi_{\mathbf{q}}(\mathbf{z}_{10}) \\ t_{10} = \pi_{t_{10}}(\mathbf{z}_{10}) \\ y_{10} = f(\mathbf{z}_{10}) \end{array}\right. \tag{6}$$

où $\pi_v$ 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.

Dans ce contexte, en prenant comme nœud la variable $\mathbf{q}$, l'algorithme de propagation-rétropropagation s'écrit comme indiqué sur le tableau 3. 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 $\mathbb{R}^3$ représentant les domaines pour les $\mathbf{z}_i$. Les pas 1 à 7 représentent la propagation et les pas 8 à 14 forment la rétropropagation.

| | |
|---|---|
| entrées : | $[p_1], [p_2], [t_1], \ldots [t_{10}], [y_1], \ldots [y_{10}]$ |
| 1 | $\mathbb{Q} := \pi_{p_1}^{-1}([p_1])$ ; |
| 2 | $\mathbb{Q} := \mathbb{Q} \cap \pi_{p_2}^{-1}([p_2])$ ; // début de la propagation |
| 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 // fin de la propagation |
| 8 | for $i := 1$ to $10$ // début de la rétropropagation |
| 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 | $[p_2] := [p_2] \cap \pi_{p_2}(\mathbb{Q})$ ; // fin de la rétropropagation |
| sorties : | $[p_1], [p_2], [t_1], \ldots [t_{10}], [y_1], \ldots [y_{10}]$ |

**Tableau 3**. *A*lgorithme 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. L'approche classique qui consiste à découper suivant toutes les directions de l'espace de recherche se traduit par un temps de calcul beaucoup trop grand pour rendre l'approche utilisable. En revanche, dans des conditions relativement peu restrictives, une méthode de regroupement se traduit par la manipulation d'ensembles de dimensions inférieures ou égales à trois (si les regroupements ont été faits convenablement) (voir [LOT 00, SAM 95]). Ceci permet d'espérer obtenir une caractérisation convenable pour les solutions du CSP en un temps très grand mais encore acceptable et ceci pour une grande classe de CSP.

La grande classe des CSP dont nous venons de parler comprend les CSP décomposables en contraintes binaires ou ternaires qui sont convexes par rapport aux intervalles. C'est le cas par exemple des CSP formés de contraintes du type $f(\mathbf{x}) = 0$, où $f$ ne comprend que des additions, des multiplications et des fonctions croissantes (exp, log, etc). 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

Cet article 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 où les inconnues sont à la fois les valeurs des paramètres inconnus, les temps de mesure incertains et les valeurs des sorties mesurées, elles aussi incertaines ("incertain" signifiant ici : "inconnu mais borné"). De plus, cet article montre les liens étroits qui existent 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 existant pour les CSP ne demandent qu'à être appliqués au contexte de l'estimation à erreurs bornées.

Toutefois, l'utilisation des techniques de propagation demande un certain savoir faire et il convient d'automatiser le plus possible la démarche. L'utilisation de la notion de surcharge d'opérateur pour le calcul sur les intervalles et sur les ensembles est un premier pas dans ce sens, mais elle est loin d'être satisfaisante. L'automatisation de tâches comme la génération des contraintes primitives, où l'écriture de la procédure de rétropropagation ne peut se faire que grâce à des outils symboliques comme les analyseurs syntaxiques (Lex & Yacc par exemple). Cette étape fondamentale est actuellement en cours de développement.

NOTE. — Les algorithmes d'estimation associés à l'exemple présenté dans cet article ont été programmés en Borland C++ builder V. Les programmes ainsi que toutes les librairies associées (calcul par intervalles, graphiques, etc) sont disponibles à l'URL http ://www.istia.univ-angers.fr/~jaulin/cppJESA01.zip.

## 5. Bibliographie

[CLE 87] CLEARY J. G., « Logical arithmetic », *Future Computing Systems*, vol. 2, n° 2, 1987, p. 125-149.

[DAV 87] DAVIS E., « Constraint propagation with interval labels », *Artificial Intelligence*, vol. 32, n° 3, 1987, p. 281-331.

[DEC 89] DECHTER R., PEARL J., « Tree-Clustering for Constraint Networks », *Artificial Intelligence*, vol. 38, n° 3, 1989, p. 353–366, Elsevier.

[GRA 98] GRANVILLIERS L., « Consistances locales et transformations symboliques de contraintes d'intervalles », PhD dissertation, Université d'Orléans, France, 1998.

[HYV 92] HYVÖNEN E., « Constraint reasoning based on interval arithmetic ; The tolerance propagation approach », *Artificial Intelligence*, vol. 58, n° 1-3, 1992, p. 71-112.

[JAU 94] JAULIN L., « Solution globale et garantie de problèmes ensemblistes ; application à l'estimation non linéaire et à la commande robuste », PhD dissertation, Université Paris-Sud, Orsay, France, 1994, disponible sur : http ://www.istia.univ-angers.fr/~jaulin/thesejaulin.zip.

[JAU 99] JAULIN L., WALTER E., « Guaranteed Bounded-Error Parameter Estimation for Nonlinear Models with Uncertain Experimental Factors », *Automatica*, vol. 35, n° 5, 1999, p. 849-856.

[JAU 00a] JAULIN L., « Interval Constraint Propagation with Application to Bounded-Error estimation », *Automatica*, vol. 36, n° 10, 2000, p. 1547-1552.

[JAU 00b] JAULIN L., « Le calcul ensembliste par analyse par intervalles », Habilitation à diriger des recherches, Université Paris-Sud, Orsay, France, 2000, disponible sur : http ://www.istia.univ-angers.fr/~jaulin/hdrjaulin.zip.

[JAU 01a] JAULIN L., KIEFFER M., BRAEMS I., WALTER E., « Guaranteed Nonlinear Estimation Using Constraint Propagation on Sets », *International Journal of Control*, vol. 74, n° 18, 2001, p. 1772-1782.

[JAU 01b] JAULIN L., KIEFFER M., DIDRIT O., WALTER E., *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*, Springer-Verlag, Londre, 2001.

[KIE 99] KIEFFER M., « Estimation ensembliste par analyse par intervalles, application à la localisation d'un véhicule », PhD dissertation, Université Paris-Sud, Orsay, France, 1999.

[LHO 97] LHOMME O., RUEHER M., « Application des techniques CSP au raisonnement sur les intervalles », *Revue d'Intelligence Artificielle*, vol. 11, n° 3, 1997, p. 283-311.

[LOT 00] LOTTAZ C., « Collaborative design using solution spaces », PhD dissertation 2119, Swiss Federal Institute of Technology in Lausanne, Suisse, 2000.

[SAM 95] SAM-HAROUD D., « Constraint consistency techniques for continuous domains », PhD dissertation 1423, Swiss Federal Institute of Technology in Lausanne, Suisse, 1995.

[WAL 97] WALTER E., PRONZATO L., *Identification of Parametric Models from Experimental Data*, Springer-Verlag, Londre, UK, 1997.