

Compte rendu de projet Initiation à la Recherche

MARGUERITE Alain
RINCE Romain

Université de Nantes
2 rue de la Houssinière, BP92208, F-44322 Nantes cedex 03,
FRANCE



Table des matières

1	Introduction	2
2	Présentation du problème	2
	2.1 Définition du contexte	2
	2.2 Exemple	2
3	Méthodes de calculs	3
	3.1 Méthodes formelles	3
	3.2 Méthodes numériques	3
4	Résolution par Intervalles	4
	4.1 L'Arithmétique des intervalles	4
	4.2 Utilisation des intervalles pour la notion de contraintes . . .	4
	4.3 Exemples d'application	5
5	Mise en œuvre de la méthode de résolution par intervalles	5
	5.1 Algorithmes	5
	5.2 Données en sortie	5

1 Introduction

L'objectif de ce document est de décrire les notions essentielles à retenir en ce début de projet d'initiation à la recherche. Ces notions font parties d'un même sujet d'étude, au cœurs des travaux de l'équipe OPTI. Elles concernent le sujet des contraintes et des intervalles. Aussi des notions d'optimisation seront abordées.

2 Présentation du problème

2.1 Définition du contexte

Un problème de satisfaction de contraintes (CSP) est défini par 3 éléments :

- Un ensemble de variables $\mathbf{V} = \{v_1, \dots, v_n\}$.
- Un domaine de valeurs pour chaque variable. Chaque valeur du domaine D_i associé à la variable v_i est une valeur que peut potentiellement prendre v_i : $\mathbf{D} = D_1 \times \dots \times D_n$.
- Un ensemble de contraintes (relations) \mathbf{C} restreignant les variables de \mathbf{V} défini ci-dessus : $\mathbf{C} = \{c_1, \dots, c_m\}$.

2.2 Exemple

Une représentation simple de ce type de problèmes est la recherche des intersections de deux cercles dans un plan :

Image absente

Figure 1 – Intersection de deux cercles

Dans le cadre de la recherche des intersections de deux cercles tels que : Définissons deux cercles respectivement de rayons r_1 et r_2 et de centres : (x_1, y_1) et (x_2, y_2) . On peut alors fixer les données en entrées du problème :

- Un ensemble de variables x et y telles que dans \mathbb{R} .
- Un domaine de valeurs pour chaque variable. La valeur du rayon de chaque cercle est située dans \mathbb{R}^+ . Les coordonnées sont elles situées dans \mathbb{R} .
- L'ensemble des contrainte est composé par les équations des cercles :

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = r_1^2 \\ (x - x_2)^2 + (y - y_2)^2 = r_2^2 \end{cases} \quad (1)$$

3 Méthodes de calculs

Les méthodes de calcul sont les solutions apportées pour permettre la résolution de problèmes mathématiques en machine ; en particulier lorsque l'on cherche à résoudre des problèmes sur les nombres réels. Elles permettent par exemple la résolution des CSP ou GCSP (Geometric Constraint Satisfaction Problem [C.J02]) et peuvent être divisées en deux catégories : les méthodes formelles et les méthodes numériques. .

3.1 Méthodes formelles

Les méthodes formelles permettent de résoudre des système d'équations ou d'inéquations en utilisant au maximum le calcul symbolique. Les approches les plus classiques des méthodes formelles utilisent des théories, telles que les idéaux polynomiaux pour les bases de Gröbner, ou la théorie des déterminants pour la méthode du résultant.

Ces méthodes ont l'énorme avantage de retourner des solutions exactes d'un système d'équation, ou tout du moins de minimiser l'utilisation de l'arithmétique flottante. Elles tenteront donc de résoudre le problème (1) en effectuant uniquement des opérations sur les équations sans évaluer les variables. On pourra donc avoir des expressions pour x et y représentant les solutions du problème de façon *sémantique*.

Cependant les résolutions de problèmes par des méthodes formelles sont forcément restreintes par les possibilités du calcul symbolique et ne pourront donc pas toujours offrir de solution et de mettre en œuvre des algorithmes de complexité exponentielles. Il n'existe par exemple pas de formules permettant de trouver les solutions d'un polynôme de degré supérieur ou égal à cinq.

3.2 Méthodes numériques

Les méthodes numériques consistent à évaluer de façon calculatoire la ou les solutions d'un problème. En effet elles sont capables d'essayer de résoudre n'importe quel système d'équation (ou d'égalités). Ainsi dans le cas du problème (1) la machine va affecter directement les calculs pour évaluer le résultats. Or l'utilisation de la représentation flottante et de son arithmétique pour simuler les opérations réelles va entraîner une diffusion et une augmentation de l'erreur de calcul, à tel point que l'on ne peut parfois plus assurer la validité d'une solution. On pourra d'ailleurs citer à titre d'exemple le problème de l'inversion d'une matrice mal conditionnée. Cependant ces calculs numériques utilisés par des méthodes de résolutions par intervalles permettent de contourner ces problèmes. Pour plus de

détails sur la représentation flottante, on pourra se référer à la thèse de Frédéric Goualard [F.G00]

4 Résolution par Intervalles

Les méthodes formelles et numériques, bien que performantes par certains aspects, sont rapidement limitées lorsque l'on veut résoudre des problèmes complexes ou que l'on veut pouvoir valider une solution (problème de propagation des erreurs...). C'est dans ce cadre que la méthode de résolution par intervalles a toute sa place. Construite grâce à l'arithmétique des intervalles, elle utilise aussi des notions apportées par la programmation par contrainte.

4.1 L'Arithmétique des intervalles

Cette arithmétique permet un calcul sur un ensemble \mathbb{I} d'intervalles sur \mathbb{R} . Les bornes $b1$ et $b2$ d'intervalle $[b1, b2]$ sont choisies en prenant un arrondi respectivement inférieur à $b1$ et supérieur à $b2$ de manière à garantir l'exactitude des calculs. L'extension des fonctions aux intervalles, introduite par Moore en 1966, permet une transition à des intervalles grâce à l'opérateur d'encadrement. Une liste non-exhaustive des opérations de cet opérateur est listée dans [F.G00].

4.2 Utilisation des intervalles pour la notion de contraintes

En utilisant l'arithmétique des intervalles, la méthode de résolution par intervalles utilise des notions de programmation par contraintes. On y retrouve celle de consistance. La consistance consiste à rechercher les valeurs cohérentes dans le domaine des variables pour les contraintes du CSP. Par exemple un CSP est globalement consistant lorsque toutes les valeurs des variables de son domaine appartiennent au moins à une solution. On devine qu'il peut être intéressant pour un CSP de posséder la consistance la plus forte possible. Ainsi les opérations pour la résolution de problèmes seront moins nombreuses.

Dans le cas du problème (1), il serait possible d'avoir des solutions différentes selon la consistance choisie. Avec une hull-consistance, nous obtiendrions une boîte contenant les deux solutions mais aussi une bonne partie de l'intersection des deux cercles. Tandis qu'avec une arc-consistance, nous obtiendrions deux boîtes, plus petites que dans le premier cas, et chacune contenant une des solutions.

4.3 Exemples d'application

Les performances de précision de la méthode attirent le monde industriel qui y voit à juste titre, l'occasion de diminuer ses coûts de production par exemple. Par ailleurs, la seule possibilité de trouver un minimum global intéresse régulièrement les industriels. On retrouve en effet dans [Sch03], le détails d'applications dans le secteur de la chimie industrielle mais aussi dans celui de la biologie avec une étude sur les protéines. La science fondamentale met aussi en application ces outils. C'est le cas en mathématique par exemple de problèmes tel que la conjecture de Kepler ou encore le maximum de clique.

5 Mise en œuvre de la méthode de résolution par intervalles

5.1 Algorithmes

La méthode de résolution par intervalles met en oeuvre deux opérations détaillées dans [Neu04] :

- Branch

Diviser pour mieux régner

Cette méthode consiste à diviser récursivement un problème en deux sous problèmes. Appliquée à la méthode de résolution par intervalles, on découpe en deux l'intervalle concerné (en son milieu ou non). On obtient alors deux problèmes plus «petits» à étudier.

- Prune

La découpe d'un problème en sous problèmes peut amener à une situation où un des sous problèmes créés ne contient aucune solution. On peut alors étudier et trier ces sous problèmes pour en supprimer un espaces de recherche superflu. Cette étape est appelée **pruning**.

5.2 Données en sortie

En sortie l'algorithme va nous retourner un ensemble de boîtes. Cet ensemble garantissant, à contrario des méthodes numériques classiques, que toutes les solutions y sont incluses. Il est d'ailleurs possible de savoir si tout l'espace d'une boîte est solution du problème. En contrepartie on ne peut toujours garantir l'existence d'une solution dans une enveloppe, il peut donc exister une approximation autour des résultats de l'algorithme.

Bibliographie

- [C.J02] C.Jermann. *Résolution de contraintes géométriques par rigidification récursive et propagation d'intervalles*. PhD thesis, Université de Nice-SOPHIA ANTIPOLIS UFR SCIENCES, Decembre 2002.
- [F.G00] F.Goualard. *Langage et environnements en programmation par contraintes d'intervalles*. PhD thesis, Université de Nantes, Juillet 2000.
- [Neu04] Arnold Neumaier. Complete search in continuous global optimization and constraint satisfaction. In A. Iserles, editor, *Acta Numerica*, 2004.
- [Sch03] Hermann Schichl. *Mathematical Modeling and Global Optimization*. PhD thesis, University of Vienna, November 2003.