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

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 partie d'un même sujet d'étude, au cœur des travaux de l'équipe OPTI. Elles concernent le sujet des contraintes et des intervalles.

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 $\mathbf{V} = \{v_1, ..., v_n\}$.
- 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 : $\mathbf{D} = D_1 \times ... \times D_n$.
- Un ensemble de contraintes (relations) $\mathbf{C}$ restreignant les variables de $\mathbf{V}$ défini ci-dessus : $\mathbf{C} = \{c_1, ..., cm\}$.

On s'intéressera notamment à la résolution des CSP sur le domaine continu. Dans ce cas les variables seront des intervalles à valeurs réelles et les contraintes seront généralement représentées par des équations et des inégalités.

2.2 Exemple

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

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 $\{x, y\}$.
- Un domaine de valeurs pour chaque variable.
- Les constantes du problème : les rayons de chaque cercle situés dans $\mathbb{R}^+$.
- L'ensemble des contrainte est composé par les équations des cercles :

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

3 Méthodes de calculs

Les méthodes de calculs permettent la résolution de problèmes mathématiques en machine ; en particulier lorsque l'on cherche à résoudre des problèmes sur les

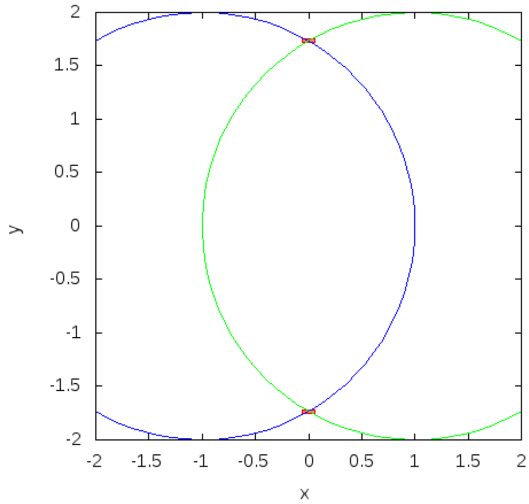

FIGURE 1 – Intersection de deux cercles

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 (2) 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 symbolique.

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 générale **et de mettre en œuvre des algorithmes de complexité exponentielles.** Il n'existe par exemple pas de formule générale 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 (2) 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 entrainer 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[Wik].

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 places. 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 l' ensemble $\mathbb{I}$ des intervalles à bornes sur $\mathbb{R}$. Les bornes $b1$ et $b2$ de l'intervalle $[b1, b2]$, résultant de tout calcul, 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 un opérateur d'encadrement. Les opérations élémentaires sont alors réécrites en terme des bornes des intervalles par exemple :

$$\begin{cases} [a..b] + [c..d] = [a + c..b + d] \\ [a..b] \times [c..d] = [min(ac, ad, bc, bd)..max(ac, ad, bc, bd)] \end{cases} \tag{2}$$

Dans le cas d'une fonction non monotone comme le cosinus, on découpe les intervalles en domaines où la fonction est monotone (2)

Une liste non-exhaustive des opérations de cet opérateur est listée dans [F.G00].

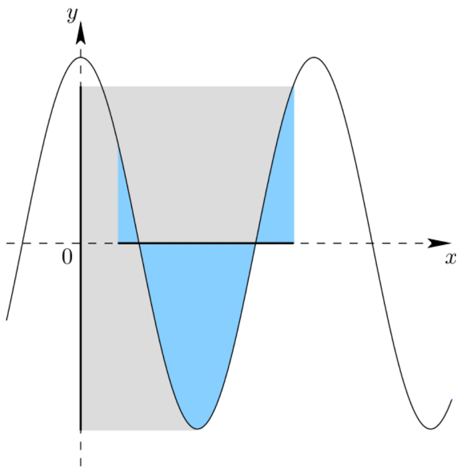

Figure 2 – Cas de la fonction cosinus

4.2 Utilisation des intervalles pour la notion de contraintes

Également, 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 (2), il serait possible d'avoir des solutions différentes selon la consistance choisie. Avec une hull-consistance, nous obtiendrions une boite contenant les deux solutions mais aussi une bonne partie de l'intersection des deux cercles. Tandis qu'avec une arc-consistance (cf [F.G00] Chapitre 3, section 3.3, définition 3.5), nous obtiendrions deux boites, plus petites que dans le premier cas, et chacune contenant une des solutions.

4.3 Exemples d'application

La précision de la méthode attirent le monde industriel. Notamment les applications composées de contraintes géométriques comme la robototique. La figure 3 illustre un robot avec deux points d'encrage (P1 et P2) possédant chacun un bras rotatif de longueurs respectives [L1] et [L2]. Ces bras sont composés de pistons permettant d'ajuster leurs longueurs. Ainsi leurs la longueur des bras sont comprises respectivement entre $[L1min..L1max]$ et $[L2min..L2max]$.

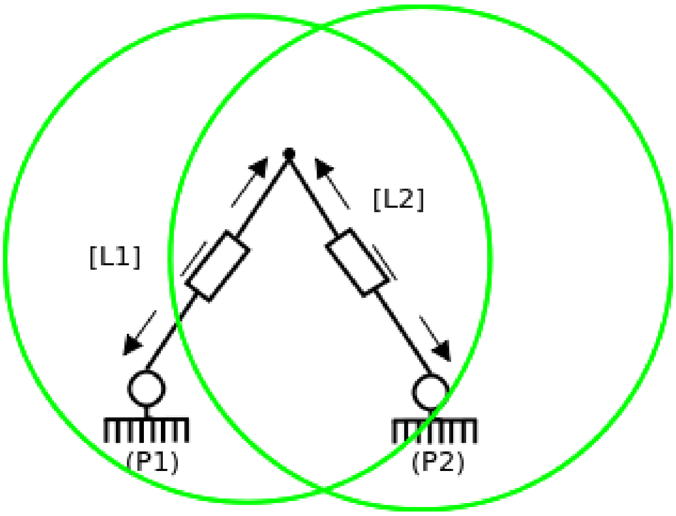

Figure 3 – Exemples d'application dans la robotique

L'objectif de ce robot est de positioner sa tête disposé au bout de ses bras sur l'intersection précise des deux cercles en vert. Les domaines de solutions de ce problème est illustré sur la figure 4 :

image manquante

Figure 4 – Illustration des domaines solutions

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

L'algorithme de Branch and Prune est au centre de la méthode de résolution par intervalles. Nous résumons ici ses deux étapes :

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.

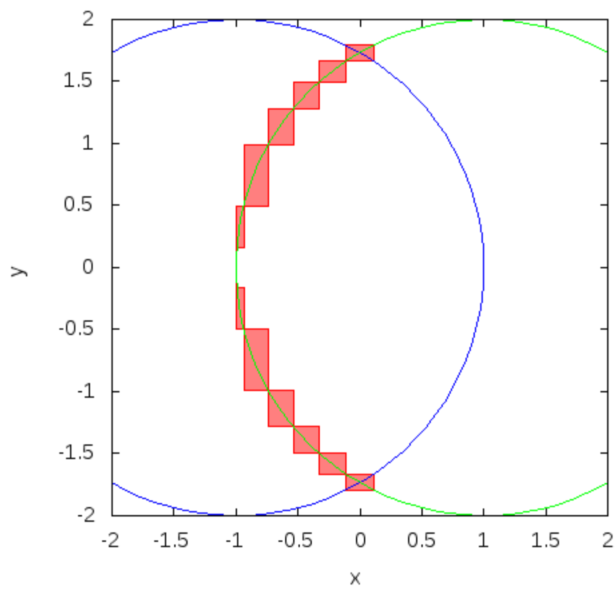

FIGURE 5 – Intersection d'un cercle et d'un disque

Prune : La découpe d'un problème en sous problèmes peut amener à une situation ou 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**. Au bout d'un certain nombre d'itération, l'algorithmes de Branch and Prune propose un ensemble de solutions sous la forme d'intervalles. On visualise sur la figure 5 un tel ensemble dans le cas d'une intersection entre un cercle et d'un disque.

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. En voici une illustration dans le cas d'une d'intersections entre deux disques :

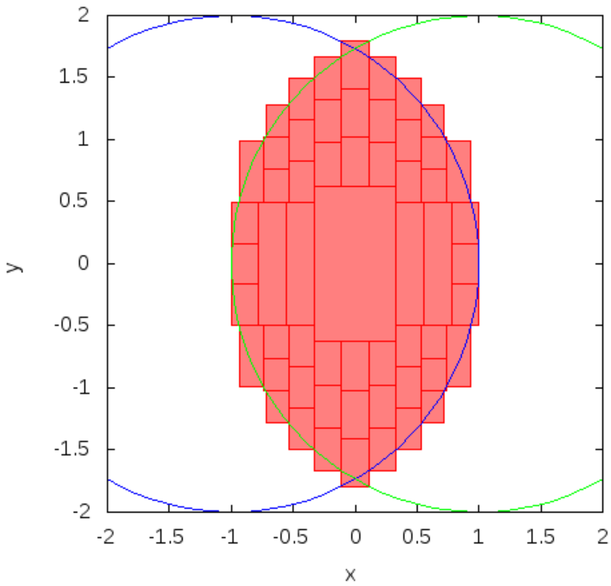

FIGURE 6 – Intersection de deux disques

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.

5.3 Outils existant

RealPaver Developpé au sein de l'équipe OPTI, cet outil reprend entre autre les méthodes de résolutions de contraintes géométriques par propagation d'intervalles. Il permet la résolution de sytèmes d'équations à n variables représentant des contraintes. Dans un soucis de précision ces solutions sont sous la forme d'intervalles. [GB06]

CHOCO Projet commun de l'École des Mines de Nantes, de LIRMM de Montpellier ansi que l'INRA de Toulouse. CHOCO est une librairie JAVA dédiée au résolution de CSP et à la programmation par contrainte. [cho99]

Google or-tools Bibliothèque Python fournissant un solveur de contraintes générique et des algorithmes : [ort11]

Bibliographie

[cho99] Librairie CHOCO , 1999.

[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.

[GB06] L. Granvilliers and F. Benhamou. Algorithm 852 : Realpaver : an Interval Solver using Constraint Satisfaction Techniques, 2006.

[ort11] Google or-tools, 2011.

[Sch03] Hermann Schichl. *Mathematical Modeling and Global Optimization.* PhD thesis, University of Vienna, November 2003.

[Wik] Wikipédia. Conditionnement d'une matrice.