Les CSPs continus (ou numériques)

Julien Alexandre dit Sandretto

Department U2IS
ENSTA Paris
IA302-2020-2021

ENSTA

IP PARIS

Vu précédement

Les CSPs discrets (dont les variables peuvent prendre des valeurs discrétes dans un domaine fini) : modélisation et résolution

Nombreux problèmes sur les rééls : domaine continu

Mais CSPs travaillent sur domaines discrets (arbre de recherche)

Solution pour représenter des domaines continus finis : formalisme des **intervalles**

Introduction aux intervalles

Un exemple illustre parfaitement l'intérêt des intervalles et de leur arithmétique garantie : le polynome de Rump

$$f(a,b) = 333.75b^6 + a^2(11a^2b^2 - b^6 - 121b^4 - 2) + 5.5b^8 + a/(2b)$$

Si on l'évalue sur les entrées $(a,b) = (77617.0, 33096.0)$, qu'obtenons-nous ?

En nombres à virgule flottante : $-1.18 \cdot 10^{21}$ (matlab), $1.18 \cdot 10^{21}$ (scilab), 1.172 (multi précision) $\Rightarrow$ FAUX !

Intervalles : $f(a,b) \in [-0.8273960599469, -0.8273960599467]$

Introduction aux intervalles

Si on veut représenter $1/3$ en flottant, on aura $0.33...4$, ce qui est faux.

Ceci est dû à la taille limitée de la mantisse.

En intervalles, $1/3 \in [0.33...3, 0.33...4]$, ce qui est correct au sens de l'inclusion.

Notations

- un intervalle $[x] \in \mathbb{IR} : [x] = [\underline{x}, \overline{x}] = \{x \in \mathbb{R} \mid \underline{x} \leq x \leq \overline{x}\}$
- $\underline{x}$ la borne inférieure et $\overline{x}$ la borne supérieure
- $\mathrm{m}([x])$ le point milieu tel que $\mathrm{m}([x]) = \underline{x} + (\overline{x} - \underline{x})/2$
- $\mathrm{w}([x])$ le diamètre de $[x]$ tel que $\mathrm{w}([x]) = \overline{x} - \underline{x}$
- Une boîte (produit Cartésien) $[\mathbf{x}] \in \mathbb{IR}^n : [\mathbf{x}] = ([x_1], ..., [x_n])^T$

L'arithmétique d'intervalles

Une arithmétique est associée, elle définit les opérations classiques comme suit :

- $[\underline{x}, \overline{x}] + [\underline{y}, \overline{y}] = [\underline{x} + \underline{y}, \overline{x} + \overline{y}]$
- $[\underline{x}, \overline{x}] - [\underline{y}, \overline{y}] = [\underline{x} - \overline{y}, \overline{x} - \underline{y}]$
- $[\underline{x}, \overline{x}] * [\underline{y}, \overline{y}] =$
  $[min(\underline{x} * \underline{y}, \underline{x} * \overline{y}, \overline{x} * \underline{y}, \overline{x} * \overline{y}), max(\underline{x} * \underline{y}, \underline{x} * \overline{y}, \overline{x} * \underline{y}, \overline{x} * \overline{y})]$
- $[\underline{x}, \overline{x}]/[\underline{y}, \overline{y}] = [\underline{x}, \overline{x}] * (1/[\underline{y}, \overline{y}])$
- $1/[\underline{y}, \overline{y}] = [min(1/\underline{y}, 1/\overline{y}), max(1/\underline{y}, 1/\overline{y})]$ si $0 \notin [\underline{y}, \overline{y}]$

L'arithmétique d'intervalles

Un problème classique de cette représentation :

$$[x] - [x] = \{x - y \mid x \in [x], y \in [x]\}$$

Par exemple $[x] = [2, 3]$, et on veut calculer $[x] - [x]$. On obtient $[2, 3] - [2, 3] = [-1, 1] \neq 0$ mais $0 \in [-1, 1]$.
Arithmétique **conservative**, mais parfois **pessimiste**.
D'autres différences :

- Addition $\neq$ Soustraction^{-1}
- Multiplication $\neq$ Division^{-1}
- Multiplication est sous-distributive par l'addition : $x \times (y + z) \subset x \times y + x \times z$

Néanmoins, elle propose un avantage très puissant, elle est **toujours** correcte par inclusion.

L'arithmétique d'intervalles

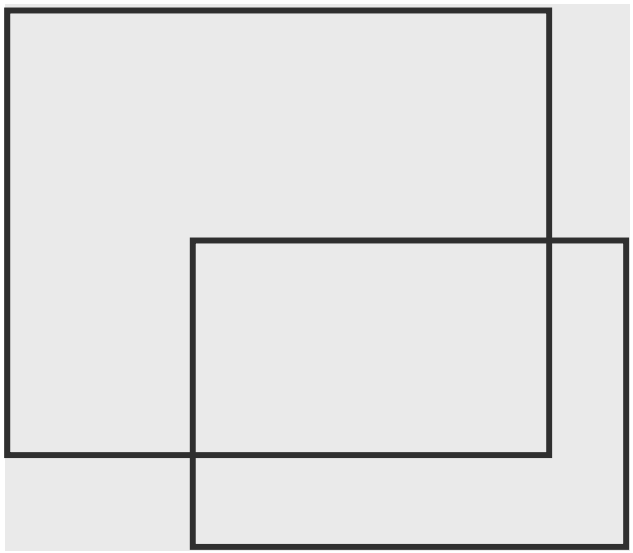
Etendre aux intervalles les fonctions élémentaires comme cos, sin, exp, acoth, log, etc.

Pour cela, monotonie de ces fonctions, ou des développements en séries, etc.

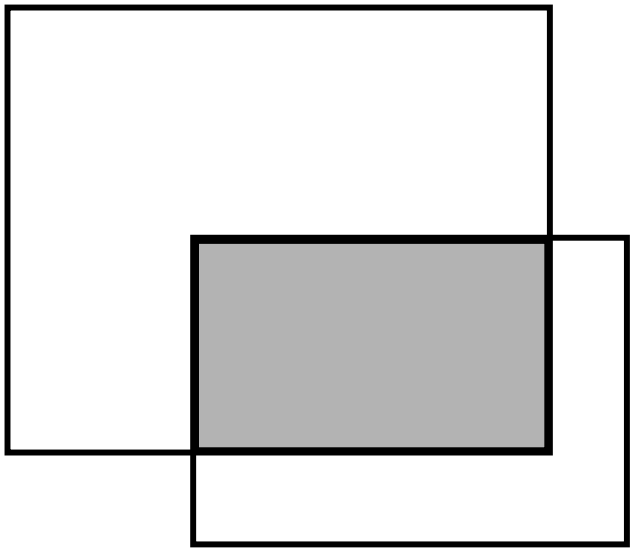
Comme les intervalles sont des ensembles, des opérations ensemblistes sont possibles comme l'union et l'intersection.

L'arithmétique d'intervalles

Union $\cup$



Intersection $\cap$



Extension aux intervalles des fonctions réelles

Etendre les fonctions réelles $f : \mathbb{R}^n \mapsto \mathbb{R}^m$ aux intervalles $[f] : \mathbb{IR}^n \mapsto \mathbb{IR}^m$, si et seulement si

$$\forall [x] \in \mathbb{IR}^n, f([x]) \subset [f]([x]).$$

On dit que $[f]$ est une *inclusion monotone* si

$$[x] \subset [y] \Rightarrow [f]([x]) \subset [f]([y])$$

Extension aux intervalles des fonctions réelles

La plus simple et directe : l'extension naturelle

Considérons la fonction $f(x) = x(x+1)$, évaluer cette fonction par l'inclusion naturelle sur l'intervalle $[x] = [-1, 1]$, donne

$$[f]_n([x]) = [x]([x] + 1) = [-1, 1] * ([-1, 1] + 1) = [-2, 2]$$

Suivant la syntaxe de l'expression : différent résultat (multiplication sous-distributive par l'addition)

$$f(x) = x * x + x = [-1, 1] * [-1, 1] + [-1, 1] = [-2, 2]$$

En utilisant la fonction élémentaire "carré" (le résultat est positif) :

$$f(x) = x^2 + x = [-1, 1]^2 + [-1, 1] = [-1, 2]$$

Extension aux intervalles des fonctions réelles

Plus évoluée : la fonction d'inclusion centrée

$$[f]_c([x]) \triangleq f(\mathrm{m}([x])) + [J_f^T]([x])([x] - \mathrm{m}([x])),$$

Où $[J_f^T]([x])$ est la fonction d'inclusion naturelle de la matrice Jacobienne de f (à l'ordre supérieur avec Hessienne $=$ Taylor)
Si nous comparons ces fonctions d'inclusion, en général :

- L'inclusion naturelle est compétitive pour les intervalles larges (diamètre supérieur à 1)
- L'inclusion de Taylor et la forme centrée sont meilleures pour les petits intervalles
- L'inclusion de Taylor est toujours meilleure que la forme centrée mais beaucoup plus chère en calcul

Extension aux intervalles des fonctions réelles

Considérons la fonction $f(x) = 4x^2 - 2x + cos(x)$ et évaluons là sur deux intervalles (un large et un petit) :

$[x] = [-1, 2.5]$: $f_N([x]) = [-15.8, 28]$ et $f_C([x]) = [-31.4, 34.5]$

$[x] = [1.1, 1.4]$: $f_N([x]) = [2.2, 6.1]$ et $f_C([x]) = [2.8, 5.3]$

Inclusion naturelle meilleure pour l'intervalle large mais que la forme centrée est meilleure pour le petit intervalle

Contraintes sur des domaines continus

L'égalité sur un ensemble ? on préfère l'appartenance

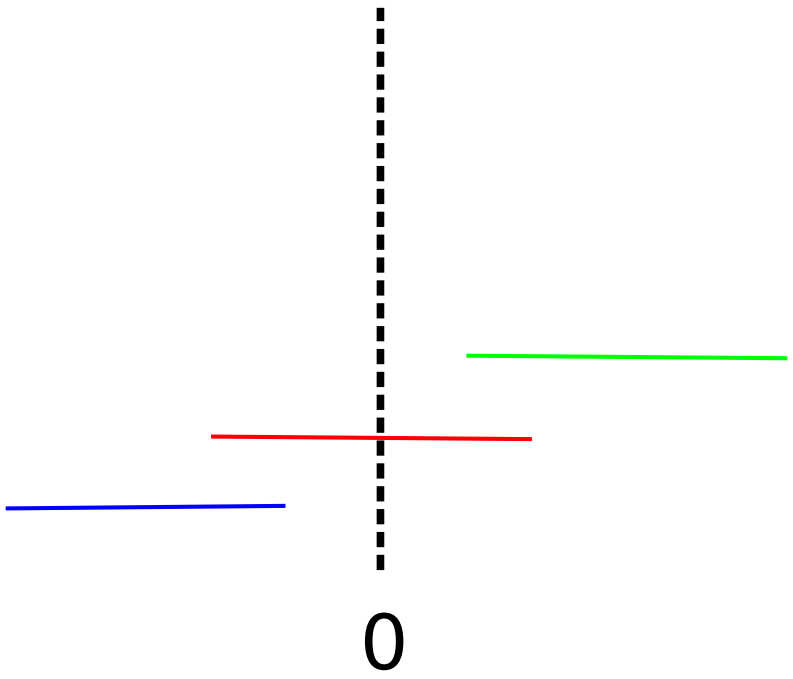
$2x + 3 = 0$, évalué sur un intervalle donne par exemple $2 * [-2, 1] + 3 = 0$, ce qui n'a que peu de sens.

Par contre, $0 \in 2x + 3$ deviendra $0 \in 2 * [-2, 1] + 3 = [-1, 5]$, ce qui est VRAI.

L'inégalité difficile à évaluer, la contrainte $[-1, 1] > 0$ est elle vrai ou fausse ? La réponse est qu'on ne sait pas...

Contraintes sur des domaines continus

- Pour savoir si un intervalle est supérieur à zéro, on regardera si sa borne inférieure est supérieure à zéro.
- Pour savoir si il est inférieur à zéro, on regardera si sa borne supérieure est inférieure à zéro.



Définition des CSPs continus

Numerical Constraint Satisfaction Problem (NCSP) : Un CSP (continu ou numérique) $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ est défini comme suit :

- $\mathcal{X} = \{x_1, \ldots, x_n\}$ est un ensemble de variables
- $\mathcal{D} = \{[d_1], \ldots, [d_n]\}$ est un ensemble de domaines $(x_i \in [d_i] \in \mathbb{IR})$
- $\mathcal{C} = \{c_1, \ldots, c_m\}$ est un ensemble de contraintes

Définition des CSPs continus

Voici un exemple de CSP continu :

$$\mathcal{H} : \begin{pmatrix} \mathcal{X} : & \{x_1, x_2, x_3, x_4\} \\ \mathcal{D} : & [\mathbf{x}] \in [-10, 10] \times [-10, 10] \times [-1, 1] \times [-1, 1] \\ \mathcal{C} : & \{x_1 + 2x_2 - x_3 = 0, \quad x_1 - x_2 - x_4 = 0\} \end{pmatrix}$$

Que veut dire résolution ?

CSPs discrets : trouver une solution (une affectation totale et consistante) ou montrer qu'il n'y en avait pas

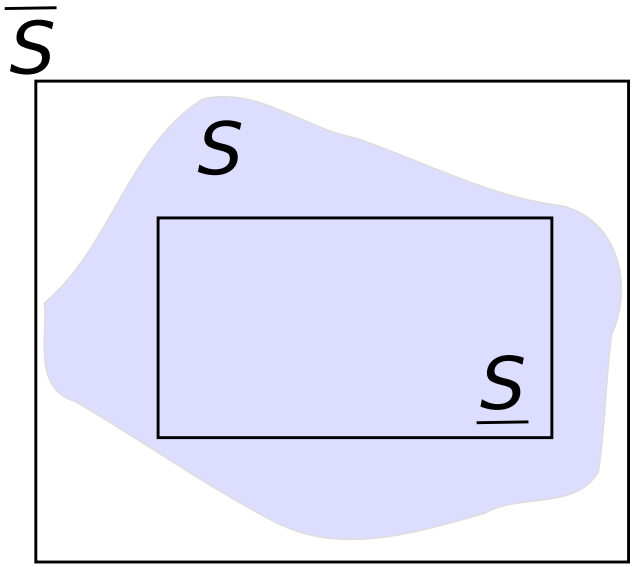
Exhiber **toutes** les solutions, puisqu'elles peuvent être énumérées

Domaine continu : ensemble de solutions continu
Comment représenter un ensemble à partir d'intervalles ?

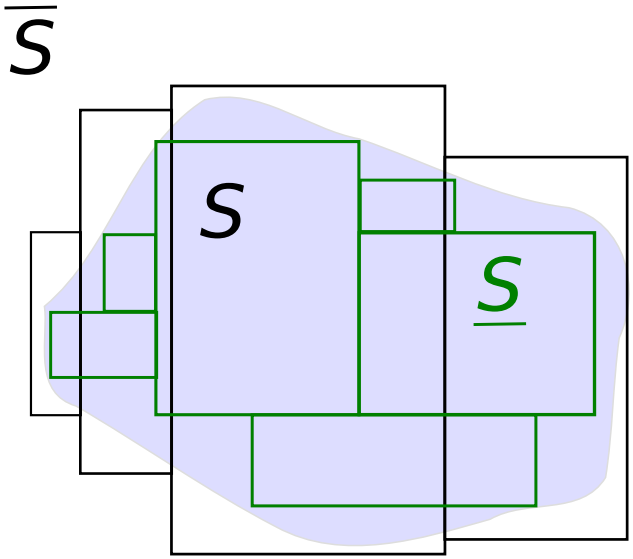
Que veut dire résolution ?

Considérons un ensemble $\mathcal{S}$, compact et fermé, représenté par deux sous-pavages (sans recouvrement) : $\underline{\mathcal{S}} \subset \mathcal{S} \subset \overline{\mathcal{S}}$.

Avec deux boîtes



Avec deux sous-pavages de N boîtes



Si on s'intéresse aux volumes : $Vol(\underline{\mathcal{S}}) < Vol(\mathcal{S}) < Vol(\overline{\mathcal{S}})$
Plus N augmente, plus $Vol(\underline{\mathcal{S}}) \nearrow$ et $Vol(\overline{\mathcal{S}}) \searrow$: meilleure qualité

Que veut dire résolution ?

Nous pouvons donc fournir plusieurs réponses à un CSP continu :

- il n'a pas de solution
- un point intérieur (une solution)
- une boîte intérieure (tous les points de cet ensemble sont solutions, mais nous n'avons pas toutes les solutions)
- une boîte extérieure (toutes les solution du CSP sont dans cet ensemble, mais certains points ne sont pas solution)
- deux sous-pavages (intérieur et extérieur)

TP : évaluation de contraintes sur des ensembles

Pour la résolution des CSP continus, nous allons utiliser Ibex.
http ://www.ibex-lib.org/doc/tutorial.html
Vous allez créer un programme en C++ pour évaluer des fonctions. En exemple, nous commencerons par évaluer les fonctions apparaissant dans le CSP vu précédement :

$$x_1 + 2x_2 - x_3 \text{ et } x_1 - x_2 - x_4$$

dans les domaines $[-10, 10], [-10, 10], [-1, 1], [-1, 1]$.
Qu'obtenez vous ? Quelles conclusions sur les contraintes d'égalité à zéro pouvez vous en tirer ?

Quelques bases d'Ibex

- Un intervalle $x_1 \in [-5, 12]$: “Interval x1(-5,12) ;”
- Arithmètique : “+,-,*,/”
- Ensemble : $\&, |, \& =, | =$
- Pour les cours suivants :
  “Variable x(2) ;”
  “Function f(x, x[0]+x[1]) ;”
  “NumConstraint csp(f, EQ) ;”
  ”CtcFwdBwd ctc(csp) ;“
  ”ctc.contract(x) ;“

Solution évaluation de fonction en Ibex

Le code très simple suivant peut être suffisant :

```
Interval x1(-10,10);
Interval x2(-10,10);
Interval x3(-1,1);
Interval x4(-1,1);



std::cout << x1+2*x2-x3 << std::endl;
std::cout << x1-x2-x4 << std::endl;
```

On obtient $[-31, 31]$ et $[-21, 21]$. On voit tout de suite que l'égalité à zéro pose problème, que pouvons nous répondre, oui ou non ? En fait, on devrait regarder si zéro appartient aux ensembles évalués. Ici c'est le cas, on sait donc qu'il y a potentiellement une solution.

TP : évaluation de contraintes sur des ensembles

Remplacez le domaine de x_4 par $[21, 22]$. Qu'obtenez vous ? Qu'en déduisez vous ?

Solution évaluation de fonction en Ibex

On obtient $[-31, 31]$ et $[-42, -1]$. Zéro n'appartient plus au deuxième intervalle, le CSP n'a pas de solution.

TP : évaluation de contraintes sur des ensembles

Essayez d'imaginer un algorithme de filtrage. Essayez le sur le CSP initial.

Solution évaluation de contraintes sur des ensembles

Algorithme de filtrage : Il faut réécrire les contraintes et utiliser l'intersection

```
x1 &= -2*x2 + x3;
x2 &= (x3-x1)/2.0;
x3 &= x1+2*x2;
x1 &= x2+x4;
x2 &= x1-x4;
x4 &= x1-x2;
```

Le résultat obtenu est $[-6.5, 6.5], [-5.5, 5.5], [-1, 1], [-1, 1]$.
L'évaluation des équations nous donne désormais
$[-18.5, 18.5], [-13, 13]$.

TP : évaluation de contraintes sur des ensembles

Cette algorithme de filtrage, exploitant la programmation par contraintes, est communément appellé le Forward/Backward. Imaginez comment améliorez l'efficacité de cet algorithme.

Solution évaluation de contraintes sur des ensembles

Amélioration de filtrage : utilisez un point fixe.

On atteint alors un point fixe encore loin de la solution, la bissection peut encore améliorer le résultat.

Si Ctc est le nom de notre filtre, $Ctc(A) \cup Ctc(B) \subset Ctc(A \cup B)$.