Résolution de problèmes

Marie Pelleau
marie.pelleau@unice.fr

Programmation par Contraintes

- La Programmation par Contraintes (PPC) formalise et résout des problèmes combinatoires [Montanari, 1974]
- Programmation déclarative, on spécifie le problème pas la méthode de résolution
- Résout de nombreux problèmes industriels
  - En biologie (*e.g.* structure secondaire de l'ARN [Perriquet and Barahona, 2009])
  - En logistique (*e.g.* problème d'ordonnancement [Grimes and Hebrard, 2011])
  - En vérification (*e.g.* vérification de programmes [Collavizza and Rueher, 2007], de modèles [Lazaar et al., 2012])
  - En musique [Truchet and Assayag, 2011], (*e.g.* aide à la composition [Papadopoulos et al., 2016])

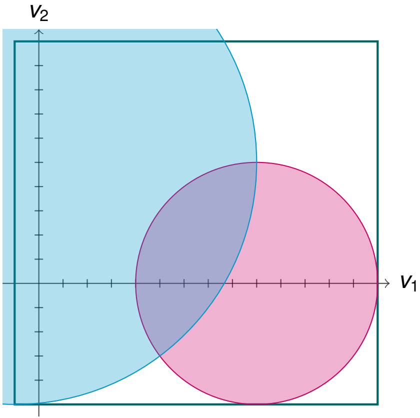
Problème de Satisfaction de Contraintes (CSP)

Définition (CSP)

- $\mathcal{V} = \{v_1, \ldots, v_n\}$: variables
- $\mathcal{D} = \{D_1, \ldots, D_n\}$: domaines
- $\mathcal{C} = \{C_1, \ldots, C_p\}$: contraintes

Exemple (Continu)

- $\mathcal{V} = \{v_1, v_2\}$
- $D_1 = [-1, 14], D_2 = [-5, 10]$
- $C_1 : (v_1 - 9)^2 + v_2^2 \leq 25$
- $C_2 : (v_1 + 1)^2 + (v_2 - 5)^2 \leq 100$



Problème de Satisfaction de Contraintes (CSP)

Définition (Solution exacte)

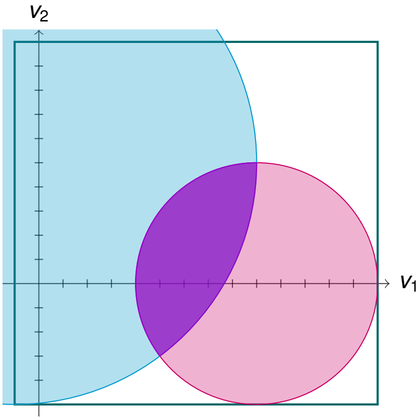
Une solution exacte est une instantiation des variables satisfaisant toutes les contraintes

Types de problème

- Trouver une solution (satisfiabilité)
- La meilleur solution (optimisation)
- Savoir si une solution existe

Remarque

Calculer les solutions exactement peut être très coûteux et même incalculable.



Coloriage de carte

Description

- 3 couleurs : bleu, rose, et vert
- 2 régions frontalières n'ont pas la même couleur

- Quelles sont les inconnues ?
  Les couleurs des régions. On a 7 variables : $\mathcal{V} = \{v_1, v_2, \ldots, v_7\}$
- Quelles sont les valeurs possibles ?
  Les couleurs. On a $D_1 = \cdots = D_7 = \{\bullet, \bullet, \bullet\}$

Coloriage de carte

Contraintes

$C_1 : v_1 \neq v_2$ $C_2 : v_1 \neq v_4$
$C_3 : v_2 \neq v_3$ $C_4 : v_2 \neq v_4$
$C_5 : v_3 \neq v_4$ $C_6 : v_3 \neq v_5$
$C_7 : v_4 \neq v_5$ $C_8 : v_4 \neq v_6$
$C_9 : v_5 \neq v_6$ $C_{10} : v_6 \neq v_7$

Send More Money

Description

$$\begin{array}{rr} & SEND \\ + & MORE \\ \hline & MONEY \end{array}$$

Chaque lettre représente un chiffre différent compris entre 0 et 9. On souhaite connaître la valeur de chaque lettre, sachant que la première lettre de chaque mot ne peut être égale à 0

- Quelles sont les inconnues ? Les lettres. On a donc 8 variables $\mathcal{V} = \{s, e, n, d, m, o, r, y\}$
- Quelles sont les valeurs possibles ? Entre 0 et 9, sauf pour s et m. On a $D_s = D_m = [1, 9], D_e = D_n = D_d = D_o = D_r = D_y = [0, 9]$

Send More Money

Description

$$\begin{array}{r} \text{S E N D} \\ + \ \ \text{M O R E} \\ \hline \text{M O N E Y} \end{array}$$

Contraintes possibles

$$C_1 : \begin{array}{rl} & s*1000 + e*100 + n*10 + d \\ & + m*1000 + o*100 + r*10 + e \\ = & m*10000 + o*1000 + n*100 + e*10 + y \end{array}$$

$$\begin{array}{lllll}
C_2 : s \neq e & C_3 : s \neq n & C_4 : s \neq d & C_5 : s \neq m & C_6 : s \neq o \\
C_7 : s \neq r & C_8 : s \neq y & C_9 : e \neq n & C_{10} : e \neq d & C_{11} : e \neq m \\
C_{12} : e \neq o & \ldots & C_{27} : o \neq r & C_{28} : o \neq y & C_{29} : r \neq y
\end{array}$$

Send More Money

Description

$$\begin{array}{rccccc} & & r_4 & r_3 & r_2 & r_1 \\ & & S & E & N & D \\ + & & M & O & R & E \\ \hline & M & O & N & E & Y \end{array}$$

Contraintes possibles

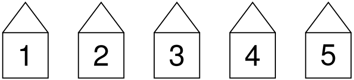
$$\begin{array}{rrcl} C_1 : & d+e &=& y+10*r_1 \\ C_2 : & r_1+n+r &=& e+10*r_2 \\ C_3 : & r_2+e+o &=& n+10*r_3 \\ C_4 : & r_3+s+m &=& o+10*r_4 \\ C_5 : & r_4 &=& m \end{array} \qquad \begin{array}{l} r_1 \in \{0,1\} \\ r_2 \in \{0,1\} \\ r_3 \in \{0,1\} \\ r_4 \in \{0,1\} \end{array}$$

$$\begin{array}{lllll} C_6 : s \neq e & C_7 : s \neq n & C_8 : s \neq d & C_9 : s \neq m & C_{10} : s \neq o \\ C_{11} : s \neq r & C_{12} : s \neq y & C_{13} : e \neq n & C_{14} : e \neq d & C_{15} : e \neq m \\ C_{16} : e \neq o & \ldots & C_{31} : o \neq r & C_{32} : o \neq y & C_{33} : r \neq y \end{array}$$

Le Zèbre

Description

Cinq maisons consécutives



- De couleurs différentes
  bleu, jaune, orange, rouge, vert
- Habitées par des hommes de différentes nationalités
  anglais, espagnol, japonais, norvégien, ukrainien
- Chacun possède un animal différent
  chien, cheval, escargot, renard, zèbre
- Chacun a une boisson préférée différente
  café, eau, lait, thé vin
- Chacun fume des cigarettes différentes
  chesterfields, cravens, gitanes, kools, old golds

Le Zèbre

Description

1. Le norvégien habite la première maison
2. La maison à coté de celle du norvégien est bleue
3. L'habitant de la troisième maison boit du lait
4. L'anglais habite la maison rouge
5. L'habitant de la maison verte boit du café
6. L'habitant de la maison jaune fume des kools
7. La maison orange se trouve juste après la verte
8. L'espagnol a un chien
9. L'ukrainien boit du thé
10. Le japonais fume des cravens
11. Le fumeur de old golds a un escargot
12. Le fumeur de gitanes boit du vin
13. Le voisin du fumeur de chesterfields a un renard
14. Le voisin du fumeur de kools a un cheval

Qui boit de l'eau ?
À qui appartient le zèbre ?

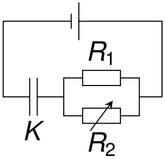
Le Zèbre

- $\mathcal{V} = \{$bleu, jaune, orange, rouge, vert, anglais, espagnol, japonais, norvégien, ukrainien, chien, cheval, escargot, renard, zèbre, café, eau, lait, thé, vin, chesterfields, cravens, gitanes, kools, old_golds$\}$
- $\forall v \in \mathcal{V},\ D_v = \{1, 2, 3, 4, 5\}$

Circuit Électrique

Desription

- Résistance R_1 de 0.1 MΩ
- Résistance variable R_2 de valeur comprise entre 0.1 MΩ et 0.4 MΩ
- Condensateur K de valeurs possibles 1 μF, 2.5 μF, 5 μF, 10 μF, 20 μF, 50 μF
- Temps de charge du condensateur soit compris entre 0.5s et 1s

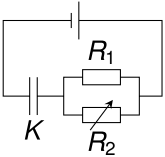


- Quelles sont les inconnues ? $\mathcal{V} = \{R_2, K, t\}$
- Quelles sont les valeurs possibles ? $D_{R_2} = [0.1 \times 10^6, 0.4 \times 10^6]$, $D_K = \{1 \times 10^{-6}, 2.5 \times 10^{-6}, 5 \times 10^{-6}, 10 \times 10^{-6}, 20 \times 10^{-6}, 50 \times 10^{-6}\}$, $D_t = [0.5, 1]$

Circuit Électrique

Desription

- Résistance R_1 de 0.1 MΩ
- Résistance variable R_2 de valeur comprise entre 0.1 MΩ et 0.4 MΩ
- Condensateur K de valeurs possibles 1 μF, 2.5 μF, 5 μF, 10 μF, 20 μF, 50 μF
- Temps de charge du condensateur soit compris entre 0.5s et 1s



Contraintes

$C_1 : \frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2}$ (loi d'ohm)

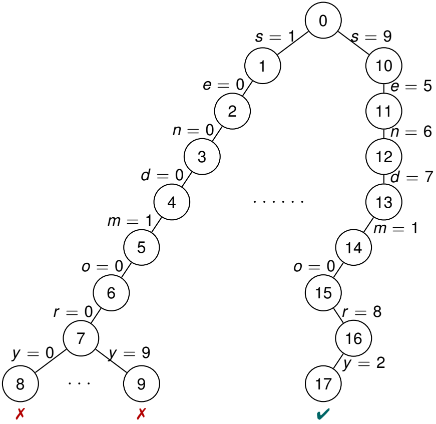
$C_2 : 1 - e^{\frac{-t}{RK}} \geq 0.99$ (loi de charge d'un condensateur)

Comment résoudre un CSP ?

Generate and Test

Méthode Naïve

Générer toutes les affectations possibles et vérifier si elles correspondent à des solutions



Remarque

Pour trouver une seule solution, on va générer :

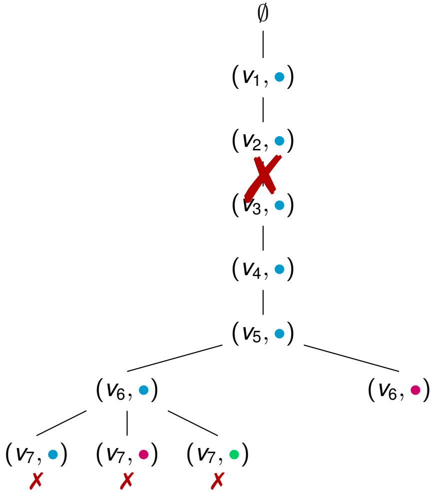
- $9^2 * 10^6 = 81\ 000\ 000$ feuilles avec la première modélisation
- $2^4 * 9^2 * 10^6 = 1\ 296\ 000\ 000$ avec la seconde

On peut faire mieux ?

Generate and Test

Coloriage de carte

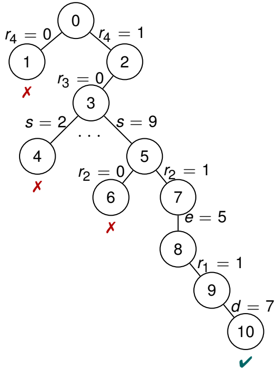
- $\mathcal{V} = \{v_1, \ldots, v_7\}$
- $D_1 = \cdots = D_7$
  $= \{\{\bullet, \bullet, \bullet\}\}$
- $C_1 : v_1 \neq v_2$
  $C_2 : v_1 \neq v_3$
  $C_3 : v_2 \neq v_3$
  $C_4 : v_2 \neq v_4$
  $C_5 : v_3 \neq v_4$
  $C_6 : v_3 \neq v_5$
  $C_7 : v_3 \neq v_6$
  $C_8 : v_4 \neq v_5$
  $C_9 : v_5 \neq v_6$
  $C_{10} : v_6 \neq v_7$



Forward Checking

Dès qu'une variable est affectée on essaye de filtrer les valeurs pour les autres variables

- On remplace la variable par sa valeur dans toutes les contraintes
- On peut filtrer si une contrainte ne contient plus qu'une variable

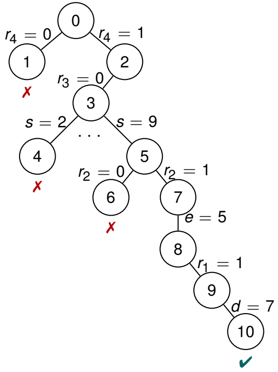


- $\{C_4 : r_3 + s + m = o + 10 * r_4,$
  $C_5 : r_4 = m\}$
  - $r_4 = 0$
    $\Rightarrow r_3 + s + m = o \wedge m = 0$ ✗
  - $r_4 = 1$
    $\Rightarrow r_3 + s + m = o + 10 \wedge m = 1$
    $\Rightarrow r_3 + s + 1 = o + 10$

Forward Checking

Dès qu'une variable est affectée on essaye de filtrer les valeurs pour les autres variables

- On remplace la variable par sa valeur dans toutes les contraintes
- On peut filtrer si une contrainte ne contient plus qu'une variable



Remarque

Pour trouver une seule solution, on va générer :

- 483 840 feuilles avec la première modélisation
- 57 avec la seconde

Pourquoi attendre une affectation ?

Méthode avec Filtrage

Les 2 étapes clés de la programmation par contraintes !

Propagation

Supprime des domaines les valeurs inconsistantes, c'est-à-dire les valeurs ne pouvant être dans une solution

Exploration

Affecte une valeur à une variable

Propagation

Consistance pour une contrainte

Différentes consistances :

- Consistance d'arc généralisée [Mackworth, 1977b]
- Consistance de chemin [Montanari, 1974]
- Consistance de borne [van Hentenryck et al., 1995]
- ...

Toutes reposent sur la notion de support

Propagation

Consistance pour une contrainte

Définition (Support)

Soient $v_1, \ldots, v_n$ des variables de domaines discrets finis $D_1, \ldots, D_n$ et C une contrainte. La valeur $x_i \in D_i$ a un support ssi $\forall j \in [1, n], j \neq i, \exists x_j \in D_j$ tel que $C(x_1, \ldots, x_n)$ soit vrai

Exemple

$C : r_4 = m$ avec $D_{r_4} = [0, 1]$ et $D_m = [1, 9]$

- 1 pour r_4 a un support : 1 pour m car $C(1, 1)$ est vrai
- 0 pour r_4 n'a pas de support : $\forall x_m \in D_m, C(0, x_m)$ est faux

Propagation

Consistance pour une contrainte

Définition (Support)

Soient $v_1, \ldots, v_n$ des variables de domaines discrets finis $D_1, \ldots, D_n$ et C une contrainte. La valeur $x_i \in D_i$ a un support ssi $\forall j \in [1, n], j \neq i, \exists x_j \in D_j$ tel que $C(x_1, \ldots, x_n)$ soit vrai

Exemple

$C : v_1 \neq v_2$ avec $D_1 = D_2 = \{\bullet, \bullet, \bullet\}$

- $\bullet$ pour v_1 a un support : $\bullet$ pour v_2 car $C(\bullet, \bullet)$ est vrai
- $\bullet$ pour v_1 a un support : $\bullet$ pour v_2 car $C(\bullet, \bullet)$ est vrai
- $\bullet$ pour v_1 a un support : $\bullet$ pour v_2 car $C(\bullet, \bullet)$ est vrai

Consistances

Définition (Consistance de bornes)

Soient $v_1, \ldots, v_n$ des variables de domaines discrets finis $D_1, \ldots, D_n$ et C une contrainte. Les domaines sont dits borne-consistants (BC) pour C ssi $\forall i \in [1, n], D_i = [a_i, b_i], a_i$ et b_i ont un support.

Exemple

Considérons deux variables v_1, v_2 de domaines $D_1 = D_2 = [-1, 4]$ et la contrainte $v_1 = 2v_2$. Les domaines borne-consistants pour cette contrainte sont $D_1 = [0, 4]$ et $D_2 = [0, 2]$

Consistances

Définition (Consistance d'arc généralisée)

Soient $v_1, \ldots, v_n$ des variables de domaines discrets finis $D_1, \ldots, D_n$ et C une contrainte. Les domaines sont dits arc-consistants généralisés (GAC) pour C ssi $\forall i \in [1, n], \forall x \in D_i, x$ a un support.

Exemple

Considérons deux variables v_1, v_2 de domaines $D_1 = D_2 = [-1, 4]$ et la contrainte $v_1 = 2v_2$. Les domaines arc-consistants pour cette contrainte sont $D_1 = \{0, 2, 4\}$ et $D_2 = \{0, 1, 2\}$

Consistance d'arc

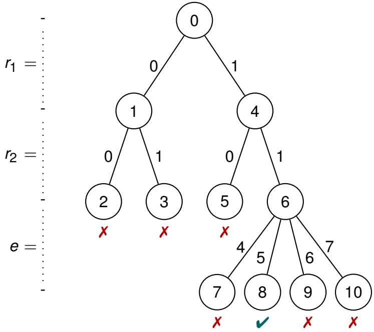
Plusieurs implémentations

- AC1 et AC3 [Mackworth, 1977a]
- AC4 [Mohr and Henderson, 1986]
- AC5 [van Hentenryck et al., 1992]
- AC6 [Bessière, 1994]
- AC7 [Bessière et al., 1999]
- AC2001 [Bessière and Régin, 2001]
- AC3.2 et AC3.3 [Lecoutre et al., 2003]

Maintaining Generalized Arc Consistency

On alterne deux phases

- Propagation, on utilise l'arc-consistance généralisée
- Exploration, on fait un choix



Remarque

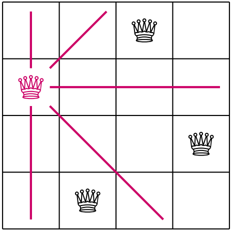
Pour trouver une seule solution, on va générer :

- 6 feuilles avec la première modélisation
- 7 avec la seconde

N-reines

Description

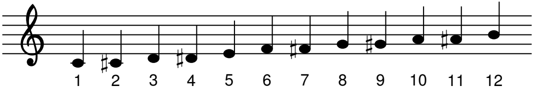
- Sur un échiquier de $n \times n$
- Placer n reines de telle sorte qu'aucune reine ne puisse en capturer une autre



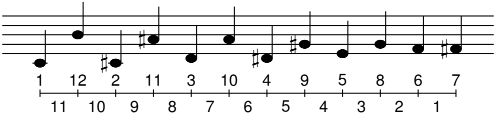
Musique : série tous intervalles

Description

- Dans les années 20, Arnold Schönberg crée un principe de composition : le dodécaphonisme
- On considère la gamme chromatique, on cherche un motif dans lequel
  - toutes les notes apparaissent exactement une fois
  - les intervalles (entre 2 notes successives) doivent être différents



Exemple (Solution triviale)



Alice et Bob vont au travail

Description

- Alice va au travail en voiture (30 à 40 min) ou par bus (au moins 60 min)
- Bob s'y rend en vélo (40 ou 50 min) ou en moto (20 à 30 min)
- Ce matin :
  - Alice a quitté sa maison entre 7h10 et 7h20
  - Bob est arrivé au travail entre 8h00 et 8h10
  - Alice est arrivée 10 à 20 min après que Bob soit parti

1. Modélisez ce problème
2. L'histoire est-elle cohérente ?
3. Quand Bob est-il parti ? Est-il possible qu'il ait pris son vélo ?
4. L'histoire est-elle cohérente si on ajoute le fait que :
   - la voiture d'Alice est en panne
   - Alice et Bob se sont rencontrés en chemin

Bessière, C. (1994).
Arc-consistency and arc-consistency again.
Artificial Intelligence, 65(1) :179–190.

Bessière, C., Freuder, E. C., and Régin, J.-C. (1999).
Using constraint metaknowledge to reduce arc consistency computation.
Artificial Intelligence, 107(1) :125–148.

Bessière, C. and Régin, J.-C. (2001).
Refining the basic constraint propagation algorithm.
In *Proceedings of the 17th International Joint Conference on Artificial intelligence (IJCAI'01)*, pages 309–315. Morgan Kaufmann.

Collavizza, H. and Rueher, M. (2007).
Exploring different constraint-based modelings for program verification.
In *Proceedings of the 13th International Conference on Principles and Practice of Constraint Programming (CP'07)*, volume 4741 of *Lecture Notes in Computer Science*, pages 49–63. Springer.

Grimes, D. and Hebrard, E. (2011).
Models and strategies for variants of the job shop scheduling problem.
In *Proceedings of the 17th International Conference on Principles and Practice of Constraint Programming (CP'11)*, volume 6876 of *Lecture Notes in Computer Science*, pages 356–372. Springer-Verlag.

Lazaar, N., Gotlieb, A., and Lebbah, Y. (2012).
A cp framework for testing cp.
Constraints, 17(2) :123–147.

Lecoutre, C., Boussemart, F., and Hemery, F. (2003).
Exploiting multidirectionality in coarse-grained arc consistency algorithms.
In *Proceedings of the 9th International Conference on Principles and Practice of Constraint Programming (CP'03)*, volume 2833 of *Lecture Notes in Computer Science*, pages 480–494. Springer.

Mackworth, A. K. (1977a).
Consistency in networks of relations.

Artificial Intelligence, 8(1) :99–118.

Mackworth, A. K. (1977b).
On reading sketch maps.
In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, pages 598–606.

Mohr, R. and Henderson, T. C. (1986).
Arc and path consistency revisited.
Artificial Intelligence, 28(2) :225–233.

Montanari, U. (1974).
Networks of constraints : Fundamental properties and applications to picture processing.
Information Science, 7(2) :95–132.

Papadopoulos, A., Roy, P., and Pachet, F. (2016).
Assisted lead sheet composition using flowcomposer.
In *Proceedings of the 22nd International Conference on Principle and Practice of Constraint Programming (CP)*.

Perriquet, O. and Barahona, P. (2009).
Constraint-based strategy for pairwise rna secondary structure prediction.
In *Proceedings of the 14th Portuguese Conference on Artificial Intelligence : Progress in Artificial Intelligence (EPIA '09)*, volume 5816 of *Lecture Notes in Computer Science*, pages 86–97. Springer-Verlag.

Truchet, C. and Assayag, G., editors (2011).
Constraint Programming in Music.
ISTE.

van Hentenryck, P., Deville, Y., and Teng, C.-M. (1992).
A generic arc-consistency algorithm and its specializations.
Artificial Intelligence, 57.

van Hentenryck, P., Saraswat, V. A., and Deville, Y. (1995).
Design, implementation, and evaluation of the constraint language cc(fd).
In *Selected Papers from Constraint Programming : Basics and Trends*, pages 293–316. Springer-Verlag.