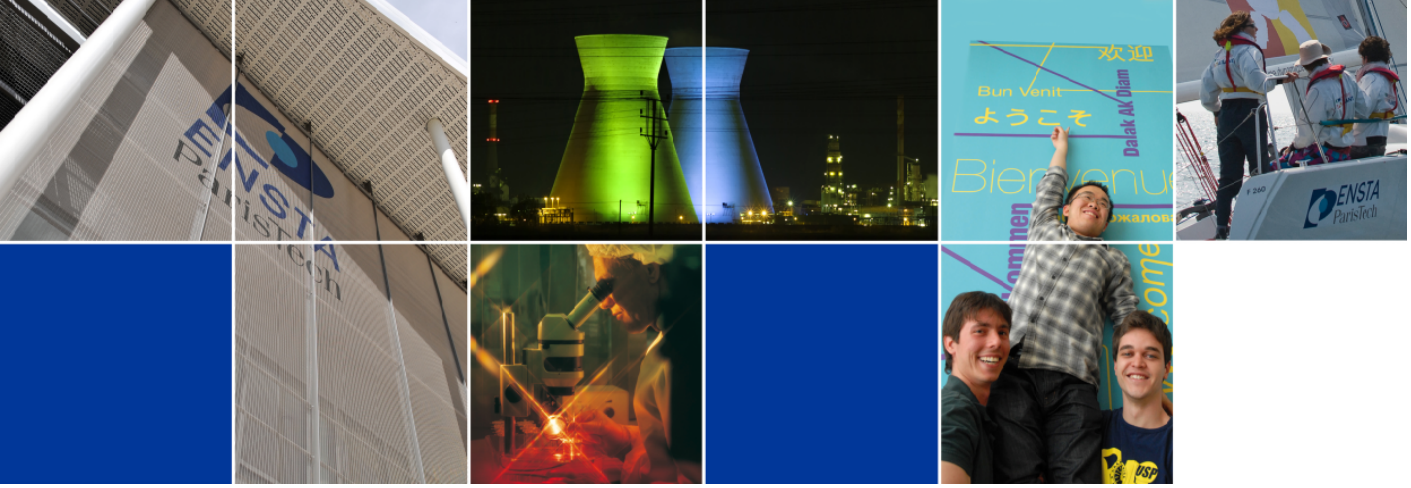

Les contraintes et les problèmes de Satisfaction de Contraintes

Julien Alexandre dit Sandretto

Department U2IS
ENSTA Paris
IA302-2020-2021

ENSTA

IP PARIS

Problèmes nécessitant un raisonnement logique

Outils mathématiques pour les représenter ou les définir:

- les contraintes;
- les problèmes de Satisfaction de Contraintes (CSP)

Exemple : Le sudoku

	4		1	3				
		3	5				1	9
					6			3
		7			5			8
	8	1				9	6	
9			2			7		
6			9					
8	1			?	2	4		
					4		9	

Exemple : Le sudoku

La règle :

- chaque case doit contenir un entier compris entre 1 et 9
- chaque entier n'apparait qu'une fois par ligne
- chaque entier n'apparait qu'une fois par colonne
- dans un carré 3×3, chaque entier n'apparait également qu'une fois

Exemple : Le sudoku

	4		1	3				
		3	5				1	9
					6			3
		7			5			8
	8	1				9	6	
9			2			7		
6			9					
8	1			?	2	4		
					4		9	

? = 1 2 3 4 5 6 7 8 9

Exemple : Le sudoku

	4		1	3				
		3	5				1	9
					6			3
		7			5			8
	8	1				9	6	
9			2			7		
6			9					
8	1			?	2	4		
					4		9	

Ligne : ? $= \not{1}\ \not{2}\ 3\ \not{4}\ 5\ 6\ 7\ \not{8}\ 9$

Exemple : Le sudoku

	4		1	3				
		3	5				1	9
					6			3
		7			5			8
	8	1				9	6	
9			2			7		
6			9					
8	1			?	2	4		
					4		9	

Colonne : ? $= \not{1}\ \not{2}\ \not{3}\ \not{4}\ 5\ 6\ 7\ \not{8}\ 9$

Exemple : Le sudoku

	4		1	3				
		3	5				1	9
					6			3
		7			5			8
	8	1				9	6	
9			2			7		
6			9					
8	1			?	2	4		
					4		9	

carré 3×3 : ? $= \not{1}\ \not{2}\ \not{3}\ \not{4}\ 5\ 6\ 7\ \not{8}\ \not{9}$

$\Rightarrow$ Ce raisonnement, par ligne, colonne puis localement peut être réalisé pour chaque case, jusqu'à la condition finale de n'avoir plus qu'une valeur possible.

Exemple : Le sudoku

Le fait de n'avoir qu'une occurence de chaque entier par ligne, colonne et par carré 3×3 fait parti de la règle du jeu, ce sont des **contraintes**. *Ces contraintes doivent être satisfaites pour résoudre le sudoku, il s'agit donc d'un problème de satisfaction de contraintes (comme beaucoup de jeux).*

Les contraintes

Definition (Contrainte)

En mathématique, une contrainte est définie comme étant une égalité ou une inégalité que doivent satisfaire les solutions d'un problème. En général, cette contrainte réduit l'espace solution. Si cette contrainte n'est pas satisfiable, elle réduit l'espace solution à l'ensemble vide.

Programmation par contraintes

Contrainte = *relation logique entre des variables*.

A ces variables sont associés des domaines qui définissent les valeurs que peuvent prendre ces variables :
$\Rightarrow$ les contraintes restreignent les valeurs des variables.

Caractéristiques

Les contraintes sont :

- relationnelles (elles lient des variables à d'autres variables),
- déclaratives (elles définissent le problème),
- permutables (l'ordre des contraintes n'est pas significatif).

Arité

Les contraintes peuvent néanmoins être classées par leur arité (le nombre de variable qu'elles contraignent), par exemple :

- **unaire** : porte sur une seule variable;
- **binaire** : porte sur deux variables;
- . . . ;
- **n-aire** : porte sur n variables (contrainte globale).

Type

Les contraintes peuvent être de différents types (suivant le domaine dans lequel le problème est posé) :

- **numériques**
  - expressions arithmétiques
  - opérations $=, \neq, <, \leq, >, \geq$
  - dans les réels, les entiers
  - opérateurs $+, -, *$,etc.
  - linéaires ou non-linéaires
- **booléennes**
  - expressions booléennes
  - opérations équivalence, non équivalence, implication
  - opérateurs et, ou, non, etc.
  - Horn, anti-Horn, etc.
- etc.

Exemples de contraintes

Numériques

- $x > 3$
- $3x^2 + 12y = 0$
- $Ax < b$

Logiques

- $p \wedge q \vee u$

Remarque : une contrainte est satisfaite si son évaluation est VRAI.

CSP

Ces contraintes définissent donc une **relation** entre des variables (mais ne disent pas comment les résoudre ou les évaluer).

Afin de représenter tout un problème (un ensemble de contraintes), nous allons utiliser le formalisme du **Problème de Satisfaction de Contraintes** (CSP pour Constraint Satisfaction Problem).

CSP

Problème modélisé sous la forme d'un ensemble de contraintes posées sur des variables, chacune de ces variables prenant ses valeurs dans un domaine.
De façon plus formelle, un triplet $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ tel que :

- $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ est l'ensemble des **variables** (les inconnues) du problème;
- $\mathcal{D} = \{D_1, D_2, \ldots, D_n\}$ associe à chaque variable x_i son **domaine** D_i, c'est-à-dire l'ensemble des valeurs que peut prendre x_i;
- $\mathcal{C} = \{c_1, c_2, \ldots, c_k\}$ est l'ensemble des **contraintes**. Chaque contrainte c_j est une relation entre certaines variables de $\mathcal{X}$, restreignant les valeurs que peuvent prendre simultanément ces variables.

Sudoku en CSP

Par exemple, si nous reprenons notre exemple du sudoku :

- $\mathcal{X}$ est l'ensemble des cases $x_{i,j}$;
- $\mathcal{D}$ associe à chaque case soit sa valeur si elle est initialisée, soit $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$;
- $\mathcal{C}$ est l'ensemble des contraintes qui définissent la régle :
  - $x_{i,j} \neq x_{i,k}, \forall k \neq j$,
  - $x_{i,j} \neq x_{k,j}, \forall k \neq i$,
  - $x_{i,j} \neq x_{k,l}, \forall (k,l) \neq (i,j)$ dans le même carré 3×3.

Sudoku

Nous avions utilisé un raisonnement par élimination, une réduction du domaine des possibles, puis un raisonnement local (sur les carrés) afin de réduire le domaine des valeurs de la case étudiée ?
$= \not{1}\ \not{2}\ \not{3}\ \not{4}\ 5\ 6\ 7\ \not{8}\ \not{9}$.

Sudoku

	4		1	3				
		3	5				1	9
					6			3
		7			5			8
	8	1				9	6	
9			2			7		
6			9					
8	1	*		?	2	4		
					4		9	

raisonnement identique : *= ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5 ~~6~~ ~~7~~ ~~8~~ 9

Cette case peut donc prendre comme valeur soit 5 soit 9.

Sudoku

	4		1	3				
		3	5				1	9
					6			3
		7			5			8
	8	1				9	6	
9			2			7		
6			9					
8	1	*	+	?	2	4	-	%
					4		9	

$+ = 3,6,7$

$- = 3,5,7$

$\% = 5,6,7.$

Sudoku

Si on résume nos réductions de domaines :

?	5	6	7
*	5	9	
+	3	6	7
-	3	5	7
%	5	6	7

Seule la case * peut et doit prendre la valeur 9.
Une contrainte implicite nous y oblige : 9 variables doivent prendre chacune une valeur différente parmi un choix de 9 implique forcément que les 9 valeurs doivent être prises.

$\Rightarrow$ Nous avons réalisé une transmission de déduction ou une **propagation**.

Programmation par contraintes

Si nous résumons notre démarche, nous avons utiliser :

- un raisonnement par élimination,
- un raisonnement local,
- puis une transmission des déductions

Il s'agit là des principes de la programmation par contraintes.

Quelques exemples d'utilisations

La programmation par contraintes est utilisée dans de nombreux domaines et pour résoudre tout un tas de problèmes divers et variés :

- les énigmes et casse-têtes en tout genre
- les problèmes "jouets" : n reines (on le verra plus tard), le sudoku ou le coloriage de cartes
- l'analyse de programme
- crypto-analyse
- la robotique
- le séquençage d'ADN
- l'ordonancement
- l'analyse grammaticale de langue étrangère
- etc.

Utilisateurs

Des compagnies utilisent la programmation par contraintes depuis les années 90 :

- Lufthansa (planification d'équipe),
- Renault (production),
- Nokia (configuration pour mobile),
- Airbus (arrangement cabine),
- la Caisse d'Epargne (gestion portefeuille)...

TP : modéliser des problèmes sous forme de CSPs

La modélisation d'un problème sous forme de CSP est complexe :

Important d'être précis (l'oubli d'une contrainte ou une erreur de domaine peut nuire au résultat obtenu par la suite).

Comment modéliser un problème sous forme de CSP ?

1. Rechercher l'ensemble des variables
2. Définir leur domaine respectif
3. Lister les contraintes définissant le problème

Rendu de monnaie

On s'intéresse à un distributeur automatique de boissons. L'utilisateur insère des pièces de monnaie pour un total de T centimes d'Euros, puis il sélectionne une boisson, dont le prix est de P centimes d'Euros (T et P étant des multiples de 10). Il s'agit alors de calculer la monnaie à rendre, sachant que le distributeur a en réserve E2 pièces de 2 euros, E1 pièces de 1 euros, C50 pièces de 50 centimes, C20 pièces de 20 centimes et C10 pièces de 10 centimes.

Comment modéliser ce problème sous forme de CSP ?

Solution : rendu de monnaie

La formulation sous forme de CSP est:

- $X = \{x_{E2}, x_{E1}, x_{C50}, x_{C20}, x_{C10}\}$, avec x_{E2} le nombre de pièces de deux euros, etc.
- $D = \{D(x_{E2}), D(x_{E1}), D(x_{C50}), D(x_{C20}), D(x_{C10})\}$, où $D(x_{E2}) = \{0, 1, \ldots, E2\}$ et idem pour les autres domaines.
- $C = \{200 * x_{E2} + 100 * x_{E1} + 50 * x_{C50} + 20 * x_{C20} + 10 * x_{C10} = T - P\}$

Ce problème a évidemment plusieurs solutions, il peut être intéressant alors de chercher la solution qui minimise le nombre de pièce rendues.

Crypto-arithmétique

Un problème de crypto-arithmétique :

$$\begin{array}{cccccc} & S & E & N & D \\ + & M & O & R & E \\ \hline = & M & O & N & E & Y \end{array}$$

Solution : Crypto-arithmétique

La formulation sous forme de CSP est:

- $X = \{S, E, N, D, M, O, R, Y\}$
- $D = \{0, \ldots, 9\}$
- $C = \{S \neq 0, M \neq 0, 1000*S+100*E+10*N+D+1000*M+100*O+10*R+E = 10000 * M + 1000 * O + 100 * N + 10 * E + Y, \text{alldifferent}\}$

La solution (mais ce n'est pas la question) est
$S = 9, E = 5, N = 6, D = 7, M = 1, O = 0, R = 8, Y = 2.$

Les condiments

Problème classique du “Salt-and-Mustard problem” proposé par Lewis Carroll.
Cinq amis, Barnabé, Casimir, Désiré, Ludovic et Martial, se retrouvent chaque jour au restaurant. Ils ont un jour décider (sans doute pour se marrer) d'appliquer les règles suivantes à chaque fois que le plat du jour est du boeuf :

- Barnabé prend du sel si et seulement si Casimir ne prend que du sel ou que de la moutarde. Il prend de la moutarde si et seulement si, ou bien Désiré ne prend ni sel ni moutarde, ou bien Martial prend les deux.
- Casimir prend du sel si et seulement si, ou bien Barnabé ne prend qu'un des deux condiments, ou bien Martial n'en prend aucun. Il prend de la moutarde si et seulement si Désiré ou Ludovic prennent les deux condiments.

Les condiments

- Désiré prend du sel si et seulement si ou bien Barnabé ne prend aucun condiment, ou bien Casimir prend les deux. Il prend de la moutarde si et seulement si Ludovic ou Martial ne prennent ni sel ni moutarde.
- Ludovic prend du sel si et seulement si Barnabé ou Désiré ne prennent ni sel ni moutarde. Il prend de la moutarde si et seulement si Casimir ou Martial ne prennent ni sel ni moutarde.
- Martial prend du sel si et seulement si Barnabé ou Ludovic prennent des deux condiments. Il prend de la moutarde si et seulement si Casimir ou Désiré ne prennent qu'un seul condiment.

Que vont-ils prendre comme condiment le jour où on leur sert un tartare de boeuf ?

Solution : Les condiments

Plusieurs approches sont possibles pour modéliser ce problème, nous choisirons celle consistant à prendre comme inconnues les choix faits par les cinqs amis. On en déduit le CSP suivant :

- $X = \{x_B, x_C, x_D, x_L, x_M\}$
- $D = \{\text{rien,sel,moutarde,les deux}\}$
- $C = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}\}$

Solution : Les condiments

avec

- c_1 = (x_B=sel ou x_B=les deux) $\Leftrightarrow$ (x_C=sel ou x_C=moutarde)
- c_2 = (x_B=moutarde ou x_B=les deux) $\Leftrightarrow$ (x_D=rien xou x_M=les deux)
- c_3 = (x_C=sel ou x_C=les deux) $\Leftrightarrow$ ((x_B=sel ou x_B=moutarde) xou x_M=rien)
- c_4 = (x_C=moutarde ou x_C=les deux) $\Leftrightarrow$ (x_D=les deux ou x_L=les deux)
- c_5 = (x_D=sel ou x_D=les deux) $\Leftrightarrow$ (x_B=rien xou x_C=les deux)
- c_6 = (x_D=moutarde ou x_D=les deux) $\Leftrightarrow$ (x_L=rien ou x_M=rien)
- c_7 = (x_L=sel ou x_L=les deux) $\Leftrightarrow$ (x_B=rien ou x_D=rien)
- c_8 = (x_L=moutarde ou x_L=les deux) $\Leftrightarrow$ (x_C=rien ou x_M=rien)
- c_9 = (x_M=sel ou x_M=les deux) $\Leftrightarrow$ (x_B=les deux ou x_L=les deux)
- c_{10} = (x_M=moutarde ou x_M=les deux) $\Leftrightarrow$ (x_C=sel ou x_C=moutarde ou x_D=sel ou x_D=moutarde).

La solution est Barnabé et Casimir prennent du sel, Désiré et Martial prennent de la moutarde, et Ludovic ne prend rien.

Les mariages stables

Une agence matrimoniale aux méthodes modernes souhaite proposer à ses clients des mariages "stables". Elle demande pour cela à chacun de ses membres de classer les membres du sexe opposé, établissant de la sorte une liste de préférences. Pour tenir compte au mieux des désirs des clients, ces listes peuvent être incomplètes, c'est-à-dire que si Paul ne veut à aucun prix être marié à Isabelle, il peut ne pas la classer (il ne la met pas dans sa liste de préférences). Par ailleurs, on peut mettre dans une liste des "ex aequo" : dans le cas où un indécis n'arrive pas à trancher entre plusieurs personnes qui lui semblent également attirantes, il peut les classer ex aequo. Pour simplifier, on supposera que l'on a autant d'hommes que de femmes.
Il s'agit alors, à partir de ces listes de préférences éventuellement incomplètes et avec des ex aequo, de former des couples stables. Par stable, on entend que personne ne soit soumis à la tentation de divorcer pour trouver mieux ailleurs : si Roméo est marié à Isabelle, et Paul à Juliette, et si à la fois Roméo préfère Juliette à Isabelle et Juliette préfère Roméo à Paul, alors ces mariages ne seront pas stables car Roméo et Juliette seront tentés de divorcer pour pouvoir convoler ensemble.

Les mariages stables

Dans l'agence matrimoniale qui nous intéresse plus particulièrement, il y a 6 hommes (que l'on appellera 1, 2, 3, 4, 5 et 6 pour préserver leur anonymat), et 6 femmes (que l'on appellera 7, 8, 9, 10, 11 et 12 pour la même raison).
Leurs préférences exprimées sont les suivantes :

Les classements des hommes :

- 1 préfère 8, puis (12 et 10 ex aequo);
- 2 préfère (8 et 11 ex aequo), puis 12;
- 3 préfère 7, puis 9, puis 12;
- 4 préfère 12, puis 9;
- 5 préfère 8, puis 7, puis 11;
- 6 préfère 12, puis (10 et 8 ex aequo), puis 11, puis 7.

Les classements des femmes :

- 7 préfère (5 et 3 ex aequo), puis 6;
- 8 préfère 2, puis, 5, puis 1, puis 6;
- 9 préfère (3 et 4 ex aequo);
- 10 préfère 6, puis 1;
- 11 préfère 5, puis 2, puis 6;
- 12 préfère 1, puis (4 et 6 ex aequo), puis 2, puis 3.

Les mariages stables

Au vu de ces préférences exprimées, les mariages de 3 avec 9 et de 6 avec 7 ne sont pas stables car 3 préfère 7 à 9 tandis que 7 préfère 3 à 6 de telle sorte que 3 et 7 seront tentés de divorcer pour se remarier ensemble. En revanche, une solution stable consisterait, par exemple, à marier 1 avec 10, 2 avec 8, 3 avec 7, 4 avec 9, 5 avec 11 et 6 avec 12.
Remarque : ce problème peut sembler quelque peu artificiel, mais si vous remplacez les hommes par des étudiants en terminal et les femmes par des universités, vous obtenez ParcoursSup.

Solution : Les mariages stables

Ici, on souhaite déterminer qui se marie avec qui. Pour exprimer cela, on peut associer à chaque homme i une variable epouse-de-i qui représente son épouse, et à chaque femme j une variable mari-de-j qui représente son mari. Néanmoins, sachant que l'on peut retrouver le mari d'une femme en cherchant l'homme dont elle est l'épouse, on peut ne garder comme variable que les épouses :

$\mathcal{X} = \{epouse - de - 1, epouse - de - 2, epouse - de - 3, epouse - de - 4, epouse - de - 5, epouse - de - 6\}$

Pour chaque variable epouse-de-i, le domaine associé contient l'ensemble des femmes qui sont classées par i et qui ont classé i :

$$\mathcal{D}(epouse - de - 1) = \{8, 10, 12\}$$
$$\mathcal{D}(epouse - de - 2) = \{8, 11, 12\}$$
$$\mathcal{D}(epouse - de - 3) = \{7, 9, 12\}$$
$$\mathcal{D}(epouse - de - 4) = \{9, 12\}$$
$$\mathcal{D}(epouse - de - 5) = \{7, 8, 11\}$$
$$\mathcal{D}(epouse - de - 6) = \{7, 8, 10, 11, 12\}$$

Solution : Les mariages stables

Enfin, les contraintes sont de deux types : $\mathcal{C} = C1, C2$
La contrainte C1 spécifie qu'il ne peut y avoir de femme mariée à plusieurs hommes :

C1 = pour tout i et tout j compris entre 1 et 6 tels que i $\neq$ j, epouse-de-i $\neq$ epouse-de-j

La contrainte C2 spécifie que les mariages doivent être stables. On rappelle que les mariages ne sont pas stables si l'on a un homme i qui préfère l'épouse de j à la sienne tandis que l'épouse de j préfère i à son mari.

Solution : Les mariages stables

Il s'agit donc dans un premier temps de formaliser cette notion de préférence. Pour cela, on peut par exemple définir le prédicat préfère de telle sorte que préfère(a,b,c) soit vrai si a préfère b à c, et faux sinon. Par exemple, les préférences de l'homme 1 peuvent être définies par : préfère(1,X,Y) $\Leftrightarrow$ (X=8 et Y=12) ou (X=8 et Y=10)

On peut ensuite définir la contrainte de stabilité des mariages par :

C2 = pour tout i et tout j compris entre 1 et 6, non (préfère(i,epouse-de-j,epouse-de-i) et préfère(epouse-de-j,i,j))

Solution : Les mariages stables

Une solution du problème des mariages stables modélisé par ce CSP est :
$\{(epouse - de - 1, 10), (epouse - de - 2, 8), (epouse - de - 3, 7), (epouse - de - 4, 9), (epouse - de - 5, 11), (epouse - de - 6, 12)\}$

Si il reste du temps

Lecture ensemble d'un article de vulgarisation : *La programmation par contraintes expliquée à ma garagiste ou à mon fleuriste*, **lemonde.fr**, 2015
Problème combinatoire, retrouvez les notions du cours, idée de résolution, applications