Introduction à la Programmation par Contraintes

Cours 1. Problèmes de Satisfaction de Contraintes.

Ruslan Sadykov

INRIA Bordeaux — Sud-Ouest

22 Septembre 2014

Organisation du cours

- 5 cours + 2 TD + 5 TP.
- 2 TD : exercices sur le papier
- 2-3 TP : introduction à un solveur PPC
- 2-3 TP : **projet**
- Évaluation : 50% de la note pour le projet + 50% de la note d'examen.
- La page du cours :

  `www.math.u-bordeaux.fr/~sadykov/teaching/MSE3315C/`

Lignes directrices

Qu'est-ce que c'est la PPC ?

Une technique mathématique pour résoudre des problèmes combinatoires

Ça marche en trois phases

Modélisation Vous définissez votre problème dans un format spécifique

Implémentation Vous écrivez votre modèle dans un langage de modélisation

Résolution Vous utiliser un solveur PPC (compatible) pour trouver une solution

Limitations de PPC

La PPC n'est **surtout pas**

- une ~~**méthode magic**~~
  - A priori,elle n'est pas meilleure que d'autres méthodes (programmation linéaire, programmation dynamique, recherches locales, etc...)
  - Tout dépend du type de problème !
- une méthode ~~« **press button** »~~, du moins pour l'instant
  - Nécessité de comprendre la technique
  - Nécessité de « guidez » la résolution

Source : Antoine Jeanjean

Les objectives du cours

- Savoir pour quels types de problèmes la méthode de PPC est bonne
- Savoir modéliser efficacement ces problèmes
- Connaître les langages de modélisation et solveurs PPC et savoir les utiliser
- Savoir comment les solveur marchent à l'intérieur

Spécificités de PPC

- On travaille avec un problème de décision — le Problème de Satisfation des Contraintes (CSP)
  (avec une série des CSPs, si c'est un problème d'optimisation)
- Grandes possibilités de modélisation
  (contraintes non-linéaires, logiques, explicites)
- Utilisation des contraintes du problème de manière active pour limiter l'espace de recherche
  (Plus il y a des contraintes, plus facile à résoudre le problème)

Un exemple introductive — Sudoku

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

- Il y a 81 cases où on peut mettre les chiffres de 1 à 9
- On doit mettre les chiffres dans les cases tel que chaque horizontale (verticale, bloque de 9 cases) contienne les chiffres différents.

On vient (presque) de formuler un Problème de Satisfaction de Contraintes(!)

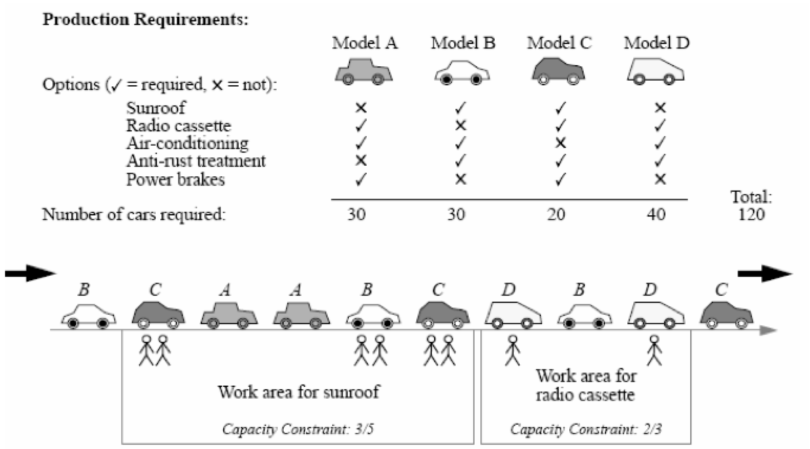
En PPC, le problème est résolu exactement plus ou moins de la même façon que vous résolvez le sudoku(!)

Lignes directrices

Domaines d'application

- Agencement
- Diagnostique et Vérification
- Planification
- Ordonnancement et Empois du temps
- Emballage et Placement
- Logistique

Application réelle I — Ligne de production des voitures

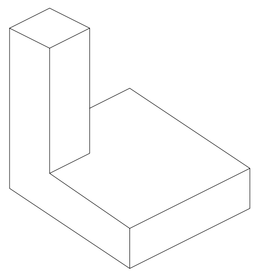


Source : Alan M. Frisch

Application réelle II — Étiquetage de scènes

On a un dessin vectoriel d'une scène contenant des objets planaires avec des sommets trièdres.
But : reconnaître ces objets, c'est-à-dire donner aux arrêtes une des 4 étiquettes possibles :

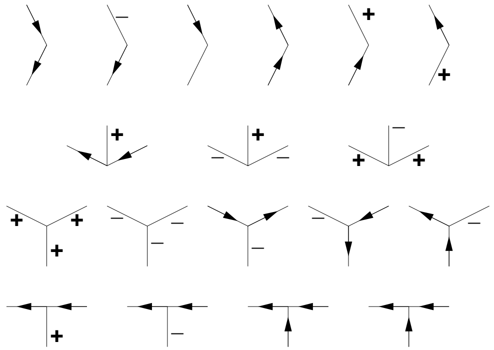
- $+$ (l'arrêt concave),
- $-$ (l'arrêt convexe),
- $\rightarrow$ ou $\leftarrow$ (l'objet à droite, l'arrière-plan à gauche).



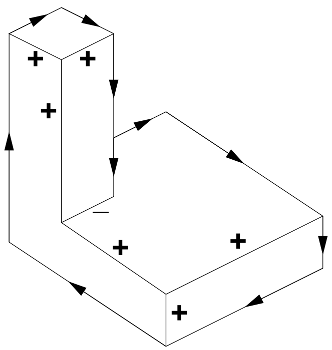
Source : (Liu, Young, Das, 88)

Application réelle II — Étiquetage de scènes (cont.)

16 jonctions possibles :



Étiquetage possible :



Application réelle III — Calendrier sportif

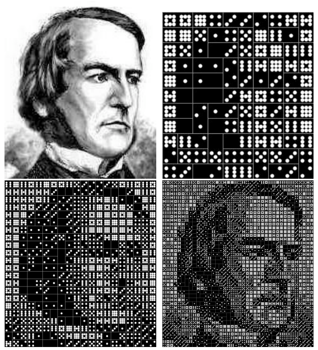
Il y a plusieurs équipes sportif. Dans le championnat, chaque équipe dois jouer avec toutes les autres. Il faut trouver un calendrier : déterminer pour chaque journée, qui joue contre qui. Des contraintes additionnelles peuvent être imposées.

Jour 1	Jour 2	Jour 3	Jour 4	Jour 5	Jour 6	Jour 7
1 vs 8	2 vs 8	4 vs 7	3 vs 6	3 vs 7	1 vs 5	2 vs 4
2 vs 3	1 vs 7	3 vs 8	5 vs 7	1 vs 4	6 vs 8	5 vs 6
4 vs 5	3 vs 5	1 vs 6	4 vs 8	2 vs 6	2 vs 7	7 vs 8
6 vs 7	4 vs 6	2 vs 5	1 vs 2	5 vs 8	3 vs 4	1 vs 3

Application plus « fun »— Portraits de dominos

But : trouver une bonne approximation d'une image réalisée à partir des dominos d'un nombre donné de boites.

Exemple à droite : Un portrait de *George Boole*, puis une séquence de portraits de dominos générés à partir de cette image et utilisant 1, 4 et 16 boites do dominos.



Source : (Cambazard, Horan, O'Mahony, O'Sullivan, 2008)

Applications réelles au Bouygues e-lab

- Les plans de tables pour le conférences du groupe
- La planification des Corps d'État Secondaires sur les chantiers de construction
- La planification de personnel
- La planification de campagnes marketing
- Projet exploitants la technique
  - La planification de la publicités (sur TF1)
  - La planification de « call-centers »

Source : Antoine Jeanjean

D'autres applications

Plus d'application sur le site

`www.csplib.org`

Il y a en a 50 !

Applications dans la recherche

Informatique	Analyse de programme, Robotique
Biologie	Repliement de protéine, Séquençage de l'ADN
Économie	Ordonnancement
Linguistique	Analyse grammaticale
Médecine	Aide à la diagnostique
Physique	Modélisation d'un système
Géographie	Systèmes de géoinformation

Lignes directrices

Problème de Satisfaction des Contraintes (CSP)

CSP est un triplé $\langle \mathbf{X}, \mathbf{D}, \mathbf{C} \rangle$, où :

- **X** est un ensemble des variables $\{x_1, \ldots, x_n\}$,
- **D** est un ensemble des domaines $\{D_{x_1}, \ldots, D_{x_n}\}$ (ensembles des valeurs possibles) pour ces variables,
- **C** est un ensemble des contraintes

$$\left\{C_i(x_{i_1}, \ldots, x_{n_i})\right\}_{i \in |C|}.$$

Chaque contrainte C_i restreint les valeurs que les variables $\{x_{i_1}, \ldots, x_{i_n}\}$ peuvent prendre simultanément.

Domaines

Les domaines peuvent être

- ensembles finis :

$$\{1, 2, \ldots, n\}, \quad \{2, 3, 5\}, \quad \{\text{rouge}, \text{noir}, \text{bleu}\};$$

- intervalles :

$$[0, k], \quad [1.2, 5.9];$$

- arbres (pas dans ce cours).

Contraintes

Les contraintes peuvent être

- logiques :

$$x = 1 \text{ ou } y = 3, \quad x = 2 \Rightarrow y = 4;$$

- arithmétiques :

$$x > y, \quad z = 2x + 3y - 5;$$

- explicites (n-tuples des valeurs possibles) :

$$(x, y) \in \{(0, 0), (1, 0), (2, 2)\}, \quad (x, y, z) \in \{(1, 2, 3), (2, 3, 4)\};$$

- complexes (globales) :

$$\mathtt{all} - \mathtt{different}(x_1, \ldots, x_n).$$

Arité des contraintes

Une contrainte peut avoir une arité quelconque :

- Une contraint est dite unaire si elle porte sur une seule variables ($x = 4$)
- Une contrainte binaire comprend deux variables ($x + y = 9$)
- Une contrainte n-aire comprend n variables

La qualificatif « n-aire » est utilisé pour une contrainte dont le nombre de variables n'est pas fixé à l'avance (par exemple, $\mathtt{all} - \mathtt{different}$)

Solutions

Solution est une affectation des valeurs $(v_1, \ldots, v_n)$ aux variables $(x_1, \ldots, x_n)$ tel que

- les valeurs sont dans les domaines : $v_j \in D_{x_j}, \forall j$;
- toutes les contraintes C_i sont satisfaites.

Un CSP est satisfiable s'il possède une solution.

Résoudre un CSP $\Leftrightarrow$ déterminer s'il est satisfiable ou pas.

Un exemple

- Les variables : x, y et z.
- Les domaines : $D_x = D_y = D_z = \{1, 2, 3\}$.
- Une contrainte : $x + y = z$.
- Les solutions : $(1, 1, 2)$, $(1, 2, 3)$, $(2, 1, 3)$.

Types de problèmes en PPC

- Trouver une solution, s'il y a une (classique).
- Trouver toutes les solutions.
- Trouver une solution qui minimise ou maximise une critère (se résout par la dichotomie).

Dichotomie

Algorithme général (minimisation)

Trouver des bornes inférieur (**BI**) et supérieur (**BS**) pour la valeur de la fonction objective;

tant que BS $-$ **BI** *est grand* **faire**

  `test` $\leftarrow$ **BI** $+ \frac{\mathbf{BS}-\mathbf{BI}}{2}$;

  si *il existe une solution* $\leq$ `test` **alors**

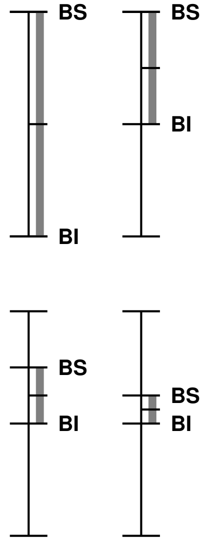
    BS $\leftarrow$ `test`;

    sauvegarder cette solution;

  sinon

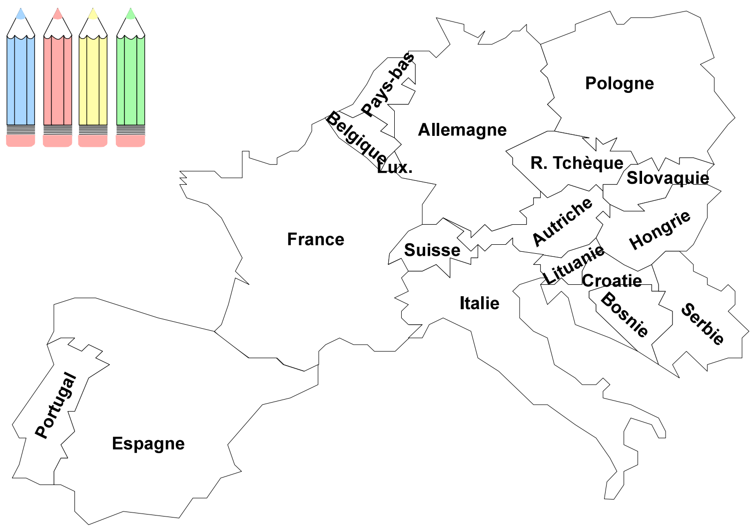
    BI $\leftarrow$ `test`;

Example



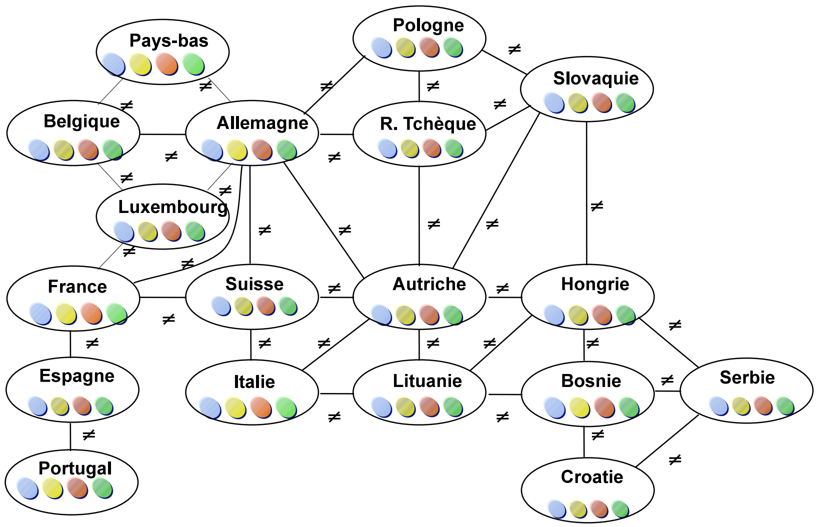
Lignes directrices

Exemple I — Coloration de carte



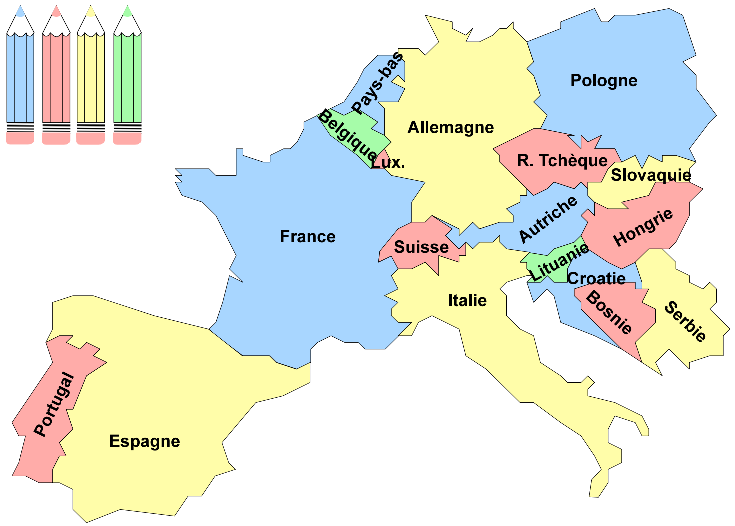
Source : Philippe Baptiste

Exemple I — Coloration de carte



Source : Philippe Baptiste

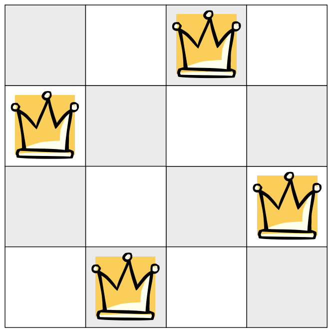
Exemple I — Coloration de carte



Source : Philippe Baptiste

Exemple II — *N* reines

Soit un échiquier de $N \times N$ cases. Placez N reines de telle sorte qu'aucune reine ne puisse en capturer une autre.



- Variables : x_i — position de reine dans la colonne i.
- Domaines : $D_{x_i} = \{1, \ldots, N\}$, $\forall i$.
- Contraintes :
  - $x_i \neq x_j$, $\forall i, j$, $1 \leq i < j \leq N$,
    $\texttt{all-different}(x_1, \ldots, x_N)$,
  - $x_i \neq x_j + (j - i)$, $1 \leq i < j \leq N$,
  - $x_i \neq x_j + (i - j)$, $1 \leq i < j \leq N$.

Exemple III — Sudoku

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

- Variables : x_{ij} — le chiffre dans le carre (i, j).
- Domaines : $D_{x_{ij}} = \{1, \ldots, 9\}$, $\forall (i, j)$.

Contraintes :

- Les chiffres dans chaque ligne sont différents :
  $\texttt{all-different}(x_{i1}, x_{i2}, \ldots, x_{i9})$, $1 \leq i \leq 9$,
- Les chiffres dans chaque colonne sont différents :
  $\texttt{all-different}(x_{1j}, x_{2j}, \ldots, x_{9j})$, $1 \leq j \leq 9$,
- Les chiffres dans chaque bloc 3×3 sont différents :
  $\texttt{all-different}(x_{3k+1,3l+1}, x_{3k+1,3l+2}, \ldots, x_{3k+3,3l+3})$,
  $0 \leq k, l \leq 2$.

Exemple IV - Retour de monnaie

On s'intéresse à un distributeur automatique de boissons. L'utilisateur insère des pièces de monnaie pour un total de T centimes, puis il sélectionne une boisson, dont le prix est de P centimes. Il faut calculer la monnaie à rendre, sachant que le distributeur a en réserve E_2 pièces de 2€, E_1 pièces de 1€, C_{50} pièces de 50 centimes, C_{20} pièces de 20 centimes et C_{10} pièces de 10 centimes.

- Variables : $x_{E2}, x_{E1}, x_{C50}, x_{C20}, x_{C10}$.
- Domaines : $D_{x_{E2}} = \{0, 1, \ldots, E_2\}$, $D_{x_{E1}} = \{0, 1, \ldots, E_1\}$,...
- Contrainte :

$$200x_{E2} + 100x_{E1} + 50x_{C50} + 20x_{C20} + 10x_{C10} = T - P$$

- Si on veut minimiser le nombre de pièces à rendre, il faut spécifier la fonction objective :

$$\min x_{E2} + x_{E1} + x_{C50} + x_{C20} + x_{C10}$$

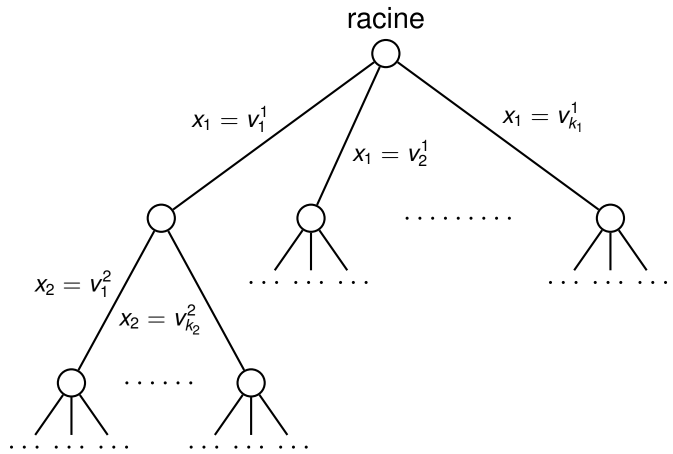
Lignes directrices

Principe de la résolution

La résolution d'un CSP ce base sur les solutions partielles et l'arbre de recherche :

- On affecte une valeur à une variable et regarde si toutes les contraintes sont encore vérifiées.
- Sinon, on « backtrack » et essaye une nouvelle valeur.
- Pour ne pas faire une énumération complète, chaque fois une valeur est affectée à une variable, les valeurs incompatibles avec cette décision dans les domaines d'autres variables sont supprimées (propagation).

L'arbre de recherche



Exemple de la propagation simple

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

$$D = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

Exemple de la propagation simple

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

$$D = \{1, 2, \cancel{3}, 4, 5, \cancel{6}, 7, \cancel{8}, \cancel{9}\}$$

Exemple de la propagation simple

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

$$D = \{1, 2, \cancel{3}, \cancel{4}, \cancel{5}, \cancel{6}, \cancel{7}, \cancel{8}, \cancel{9}\}$$

Exemple de la propagation simple

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

$$D = \{1, \cancel{2}, \cancel{3}, \cancel{4}, \cancel{5}, \cancel{6}, \cancel{7}, \cancel{8}, \cancel{9}\}$$

Exemple de la propagation avancée

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

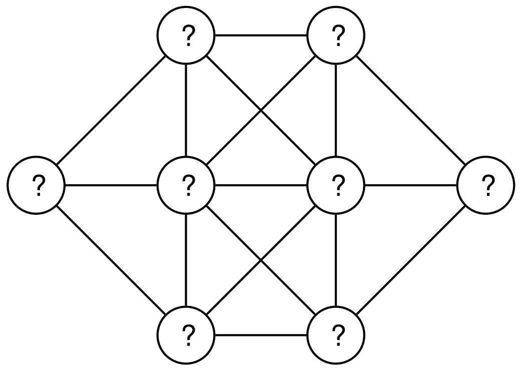
Exemple de la propagation avancée

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

Exemple de la propagation avancée

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

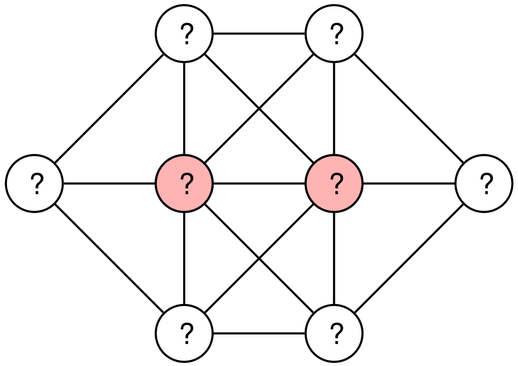
Un exemple de la résolution complète



Problème : affectez les valeurs de 1 à 8 aux sommets, chaque valeur doit apparaitre une fois, les valeurs consécutives ne doivent pas être affectées aux sommets connectés

Source : Patrick Prosser

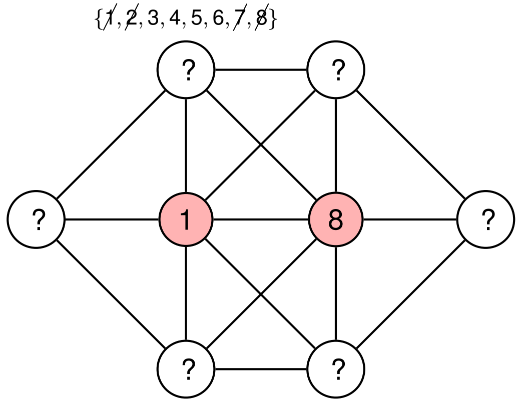
Un exemple de la résolution complète



Soyez préparés à faire le backtrack.
Quels sommets sont les plus difficiles à énumérer ?
Quelles valeurs sont les moins contraignantes ?

Source : Patrick Prosser

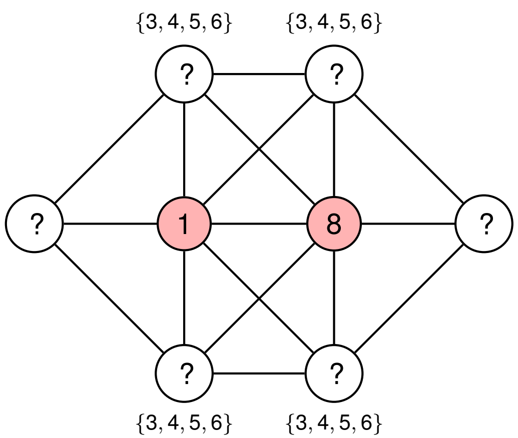
Un exemple de la résolution complète



On peux maintenant éliminer plusieurs valeurs dans les domaines d'autres sommets.

Source : Patrick Prosser

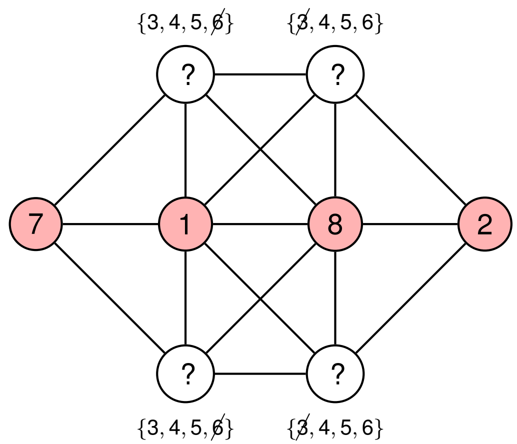
Un exemple de la résolution complète



On peux maintenant éliminer plusieurs valeurs dans les domaines d'autres sommets.

Source : Patrick Prosser

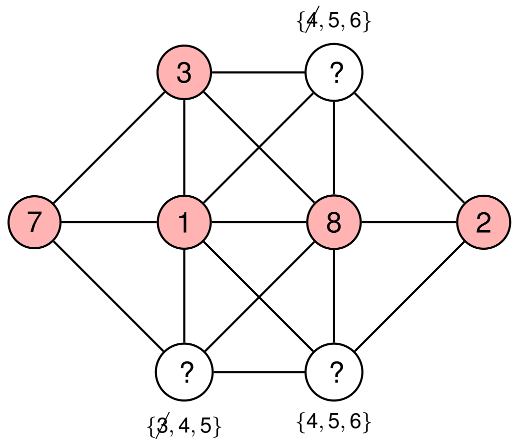
Un exemple de la résolution complète



On peux maintenant éliminer plusieurs valeurs dans les domaines d'autres sommets.

Source : Patrick Prosser

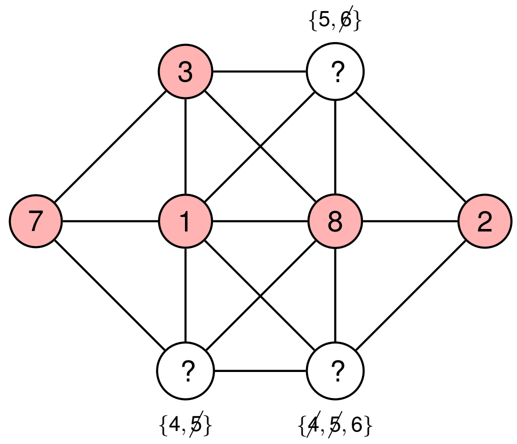
Un exemple de la résolution complète



On devine maintenant la valuer pour un sommet.
Soyez préparés à faire le backtrack.

Source : Patrick Prosser

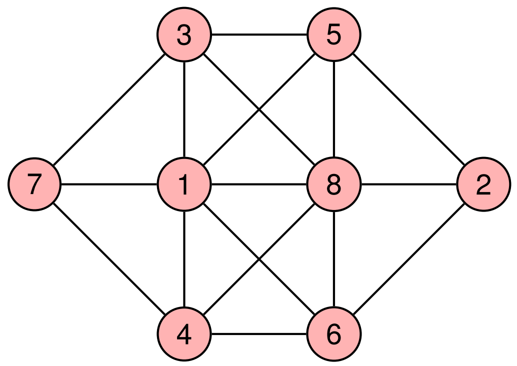
Un exemple de la résolution complète



On propage cette décision.

Source : Patrick Prosser

Un exemple de la résolution complète



Une solution.

Source : Patrick Prosser

Exemple : l'arbre de recherche