Introduction à la Programmation par Contraintes (PPC)

Ruslan Sadykov
LIX, École Polytechnique

Contenu

- Introduction
- Modélisation
  - Problèmes de satisfaction des contraintes
  - Exemples des modèles PPC simples
- Méthodes de résolution
  - Recherche arborescente
  - Consistance locale
- Les contraintes globales
- Quelques modèles PPC pratiques
- Solveurs PPC

Qu’est-ce que c’est la PPC

- Une autre façon de formuler et résoudre des problèmes combinatoires
- Spécificité de Programmation par Contraintes
  - On résout des problèmes de décision
    (dichotomie pour le problèmes d’optimisation)
  - Plus expressive que la PLNE
    (contraintes non-linéaires, logiques, explicites)
  - Utilisation des contraintes du problème de manière active pour limiter l’espace de recherche

Problèmes qu'on souvent résout avec PPC

- Ordonnancement
- Allocation des ressources
- Emplois du temps
- Conception de circuits
- Séquençage de l'ADN
- etc.

Contenu

- Introduction
- Modélisation
  - Problèmes de satisfaction des contraintes
  - Exemples des modèles PPC simples
- Méthodes de résolution
  - Recherche arborescente
  - Consistance locale
- Les contraintes globales
- Quelques modèles PPC pratiques
- Solveurs PPC

Problème de satisfaction des contraintes (CSP) – I

- Réseau de contraintes *P* est un triplé **(*X*, *D*, *C*)** avec :
  - $\boldsymbol{X}=\{X_1, X_2, \ldots, X_n\}$ – l'ensemble des variables
  - $\boldsymbol{D}=\{D_1, D_2, \ldots, D_n\}$ – l'ensemble des domaines finis
  - $\boldsymbol{C}=\{C_1, C_2, \ldots, C_e\}$ – l'ensemble des contraintes
- Chaque contrainte est un sous-ensemble du produit cartésien des domaines des variables sur lesquels elle porte.
- Les contraintes sont explicites (tuples de valeurs possibles) ou implicites (e.g. arithmétiques)

Problème de satisfaction des contraintes (CSP) – II

- Une solution est une affectation d'une valeur à chaque variable telle que les contraintes soient respectées (ou « pas de solution »)
- Un réseau de contraintes est binaire si toutes ses contraintes sont binaires (arité 2), pour chaque réseau de contraintes non-binaire, il y a un réseau binaire équivalent
- Variations : existence d'une solution, nombre de solutions, toutes les solutions

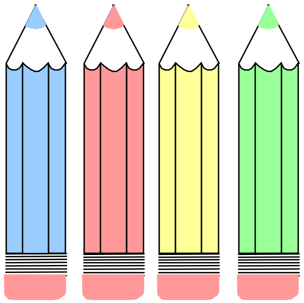
Contenu

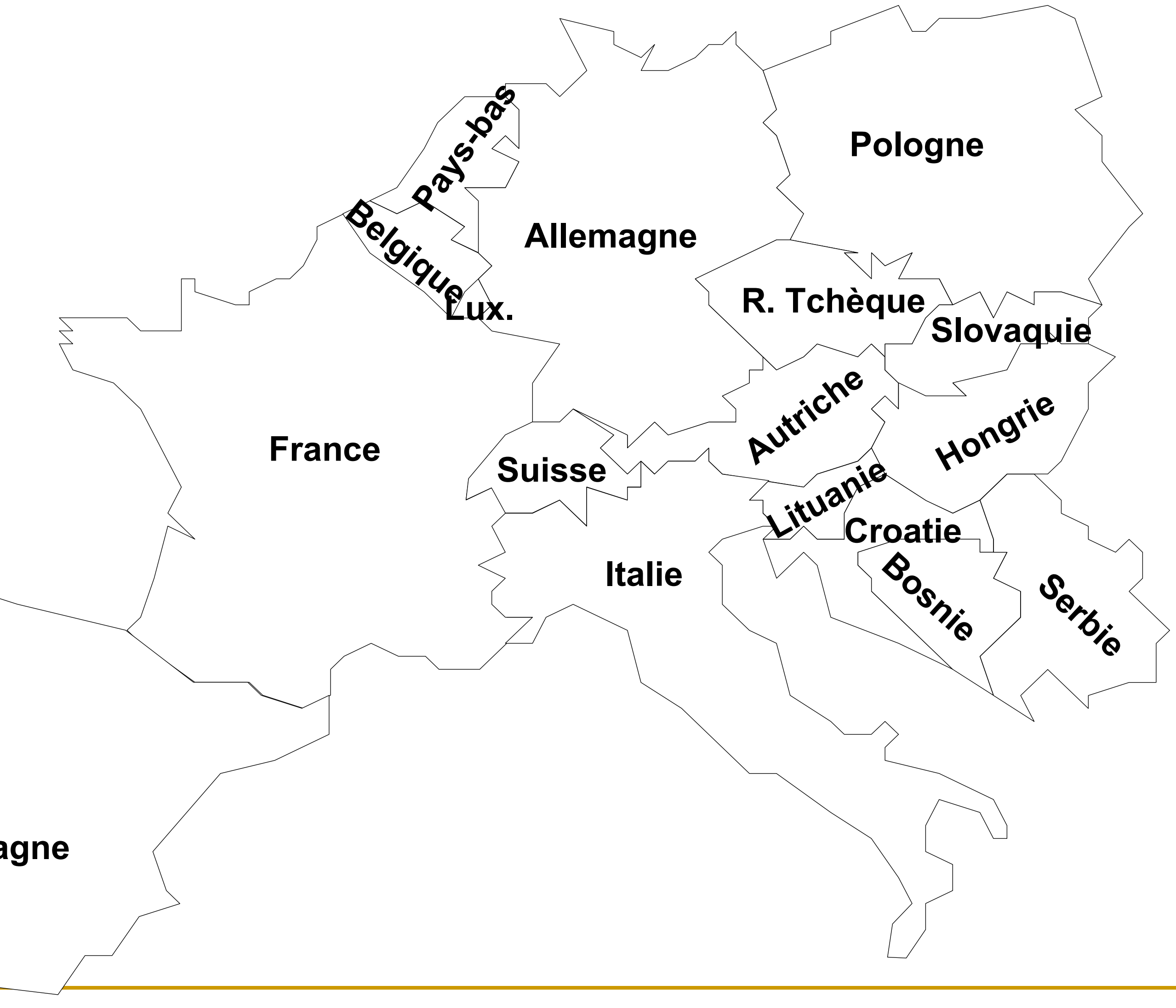
- Introduction
- Modélisation
  - Problèmes de satisfaction des contraintes
  - Exemples des modèles PPC simples
- Méthodes de résolution
  - Recherche arborescente
  - Consistance locale
- Les contraintes globales
- Quelques modèles PPC pratiques
- Solveurs PPC

Un petit exemple de CSP

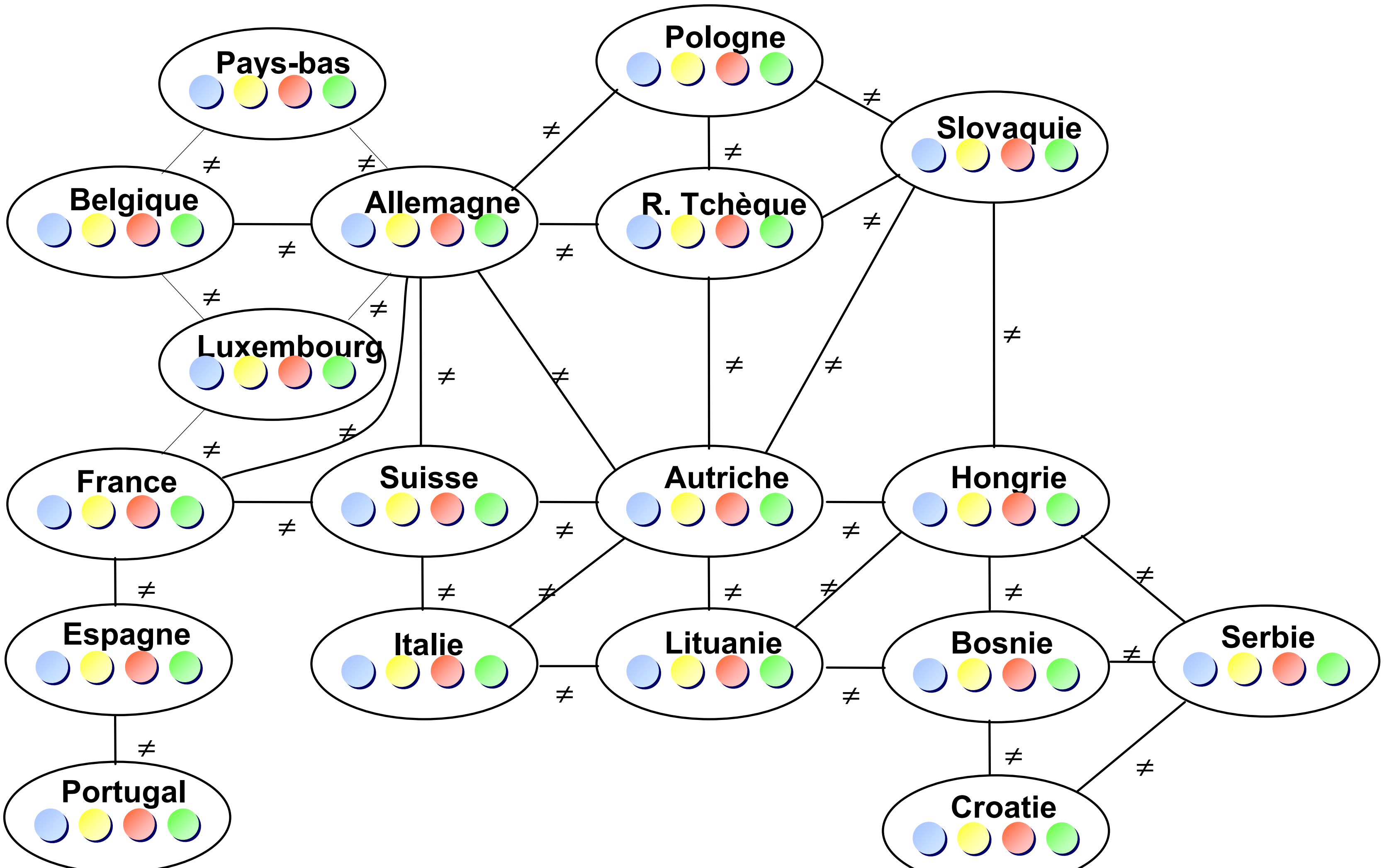
- Les variables *x*,*y* et *z* (entières).
- Les domaines $D_x=[1,3]$, $D_y=[1,3]$, $D_z=[1,3]$.
- La contrainte $x=y+z$
- Les solutions sont (2,1,1), (3,1,2), (3,2,1).

Un exemple : coloration d'une carte I

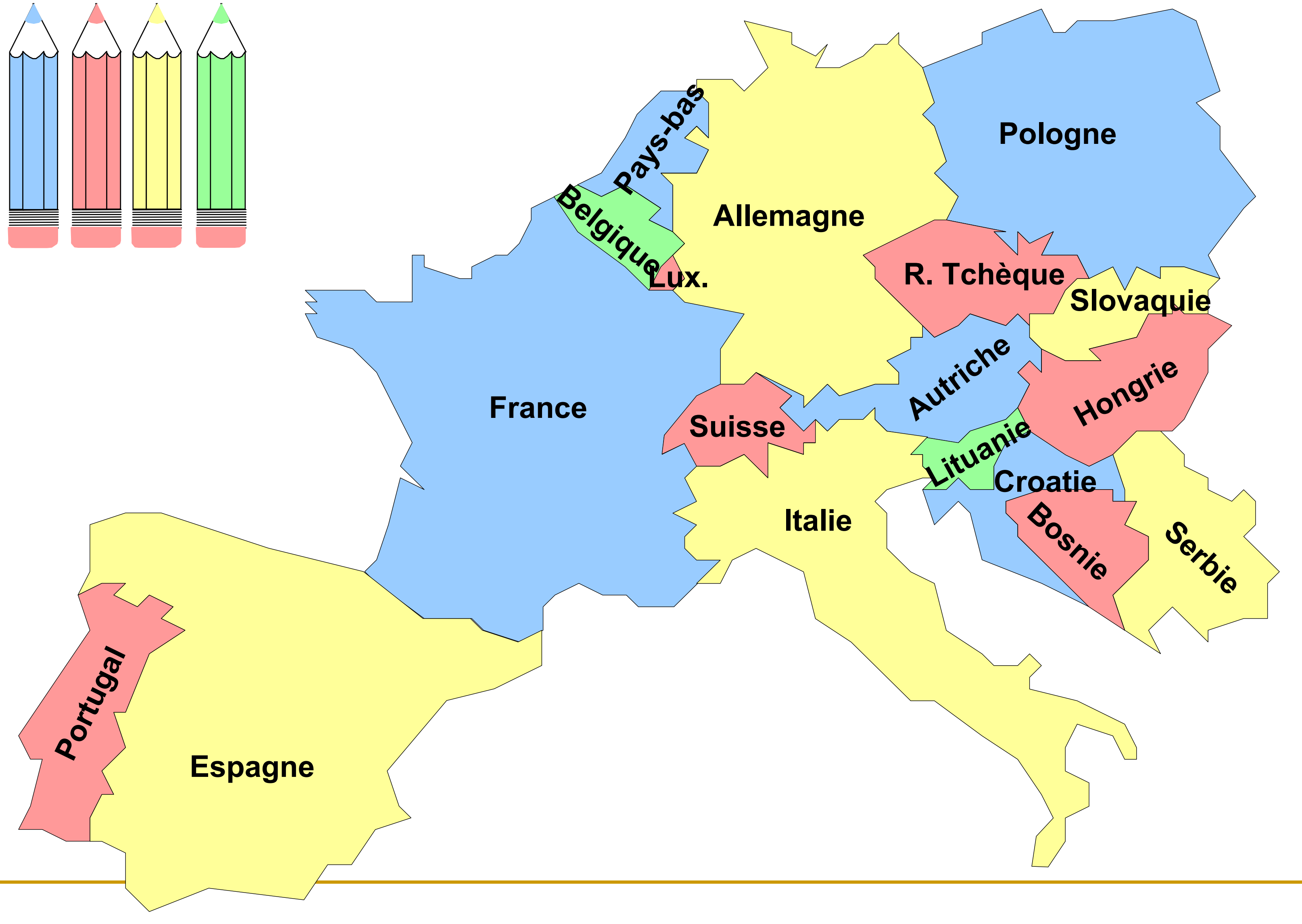




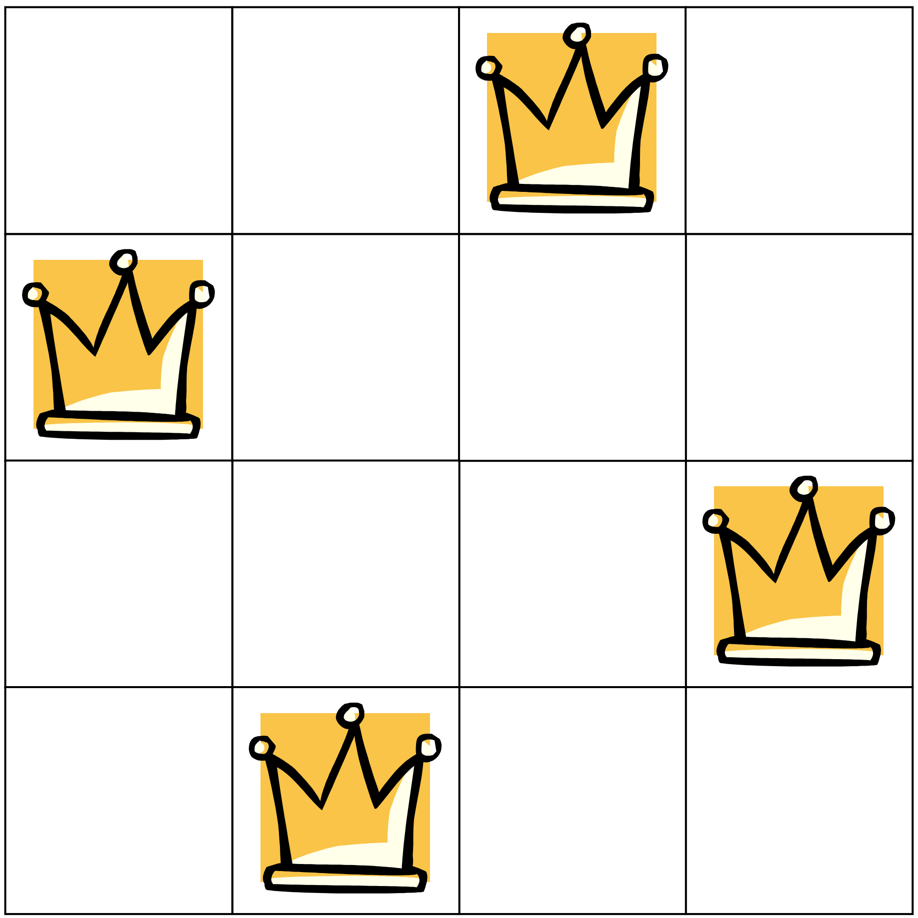
Un exemple : coloration d'une carte II



Un exemple : coloration d’une carte III



Un exemple : Les N reines



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

- Variables : X_i – position de la reine dans la colonne i
- Domaines : $D_i = \{1,2,\ldots,N\}$
- Les contraintes sont :
  - $X_i \neq X_j$, $1 \leq i < j \leq N$
  - $X_i \neq X_j+(j-i)$, $1 \leq i < j \leq N$
  - $X_i \neq X_j-(j-i)$, $1 \leq i < j \leq N$

Un exemple : carré magique

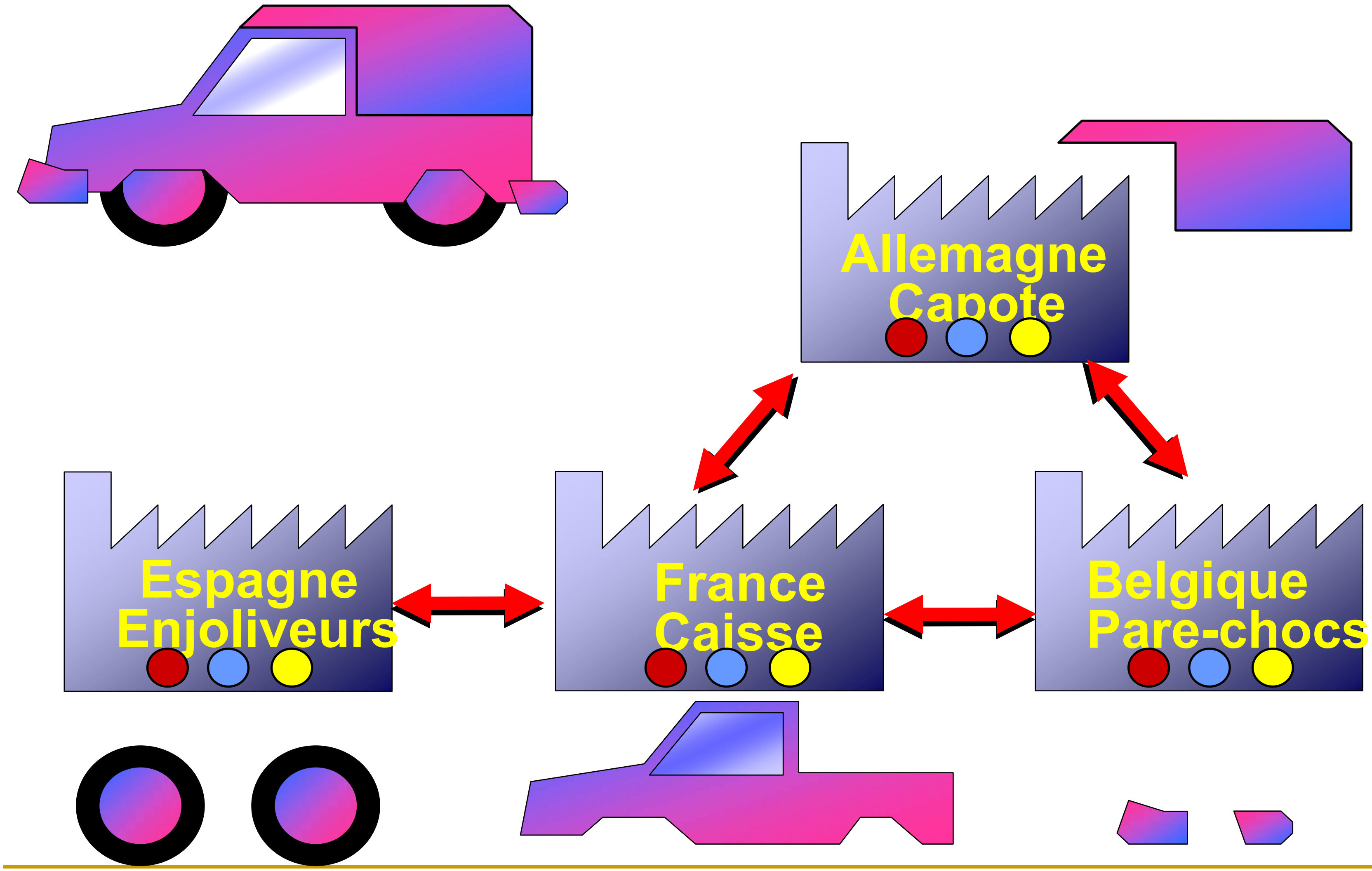
8	**1**	**6**
3	**5**	**7**
4	**9**	**2**

- Remplir une grille de $N*N$ avec les nombres de 1 à N^2 de telle sorte que toutes les lignes, toutes les colonnes et les diagonales aient la même somme
- Modèle?

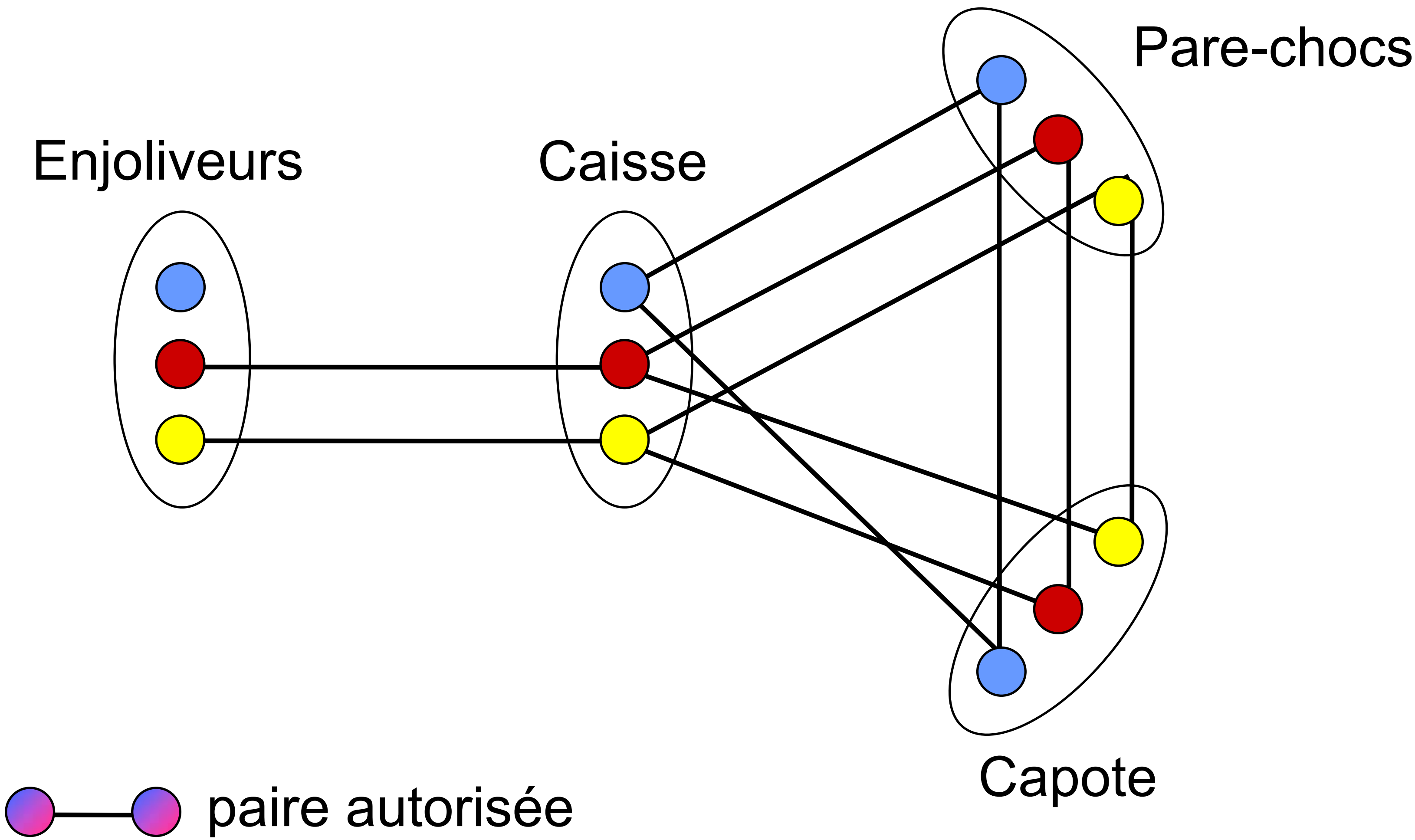
Contenu

- Introduction
- Modélisation
  - Problèmes de satisfaction des contraintes
  - Exemples des modèles PPC simples
- Méthodes de résolution
  - Recherche arborescente
  - Consistance locale
- Les contraintes globales
- Quelques modèles PPC pratiques
- Solveurs PPC

Exemple trivial



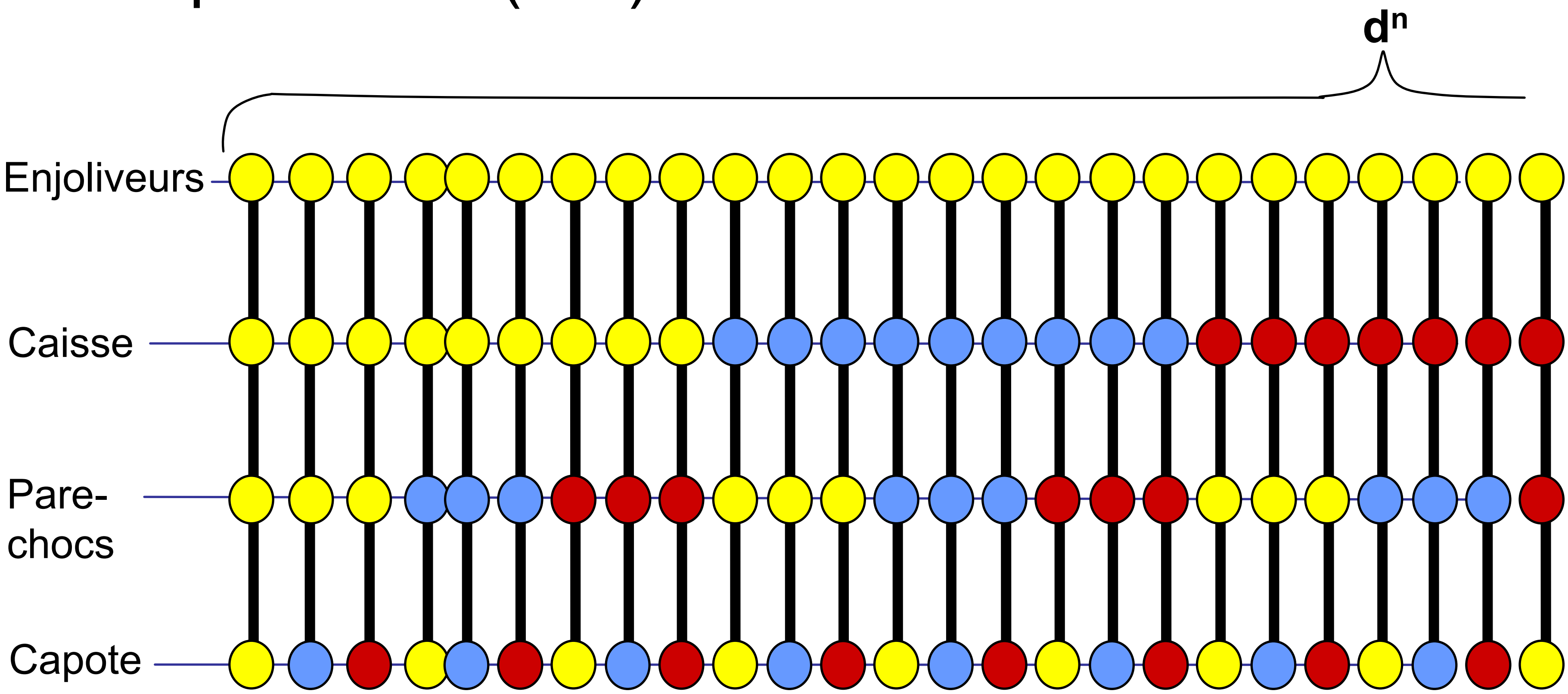
Exemple trivial : réseau de contraintes



Résolution naïve

On test toutes les combinaisons de valeurs.

Complexité : $O(ed^n)$.



Contenu

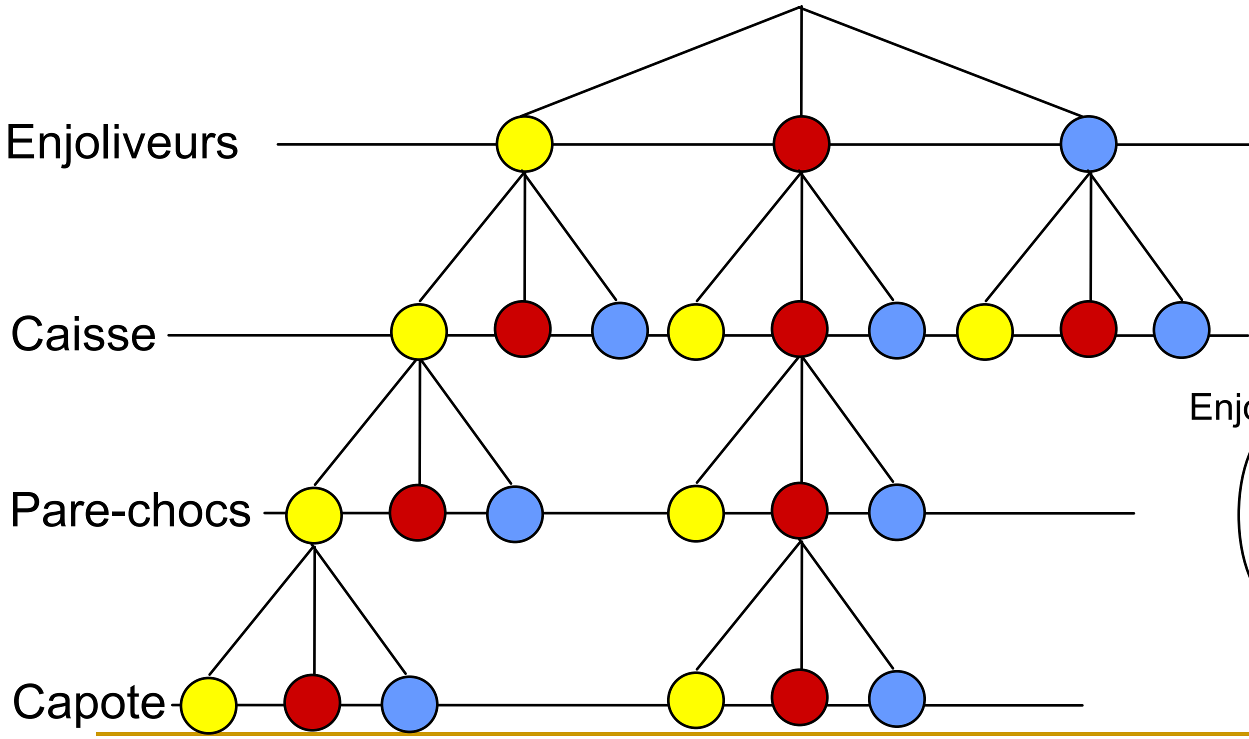
- Introduction
- Modélisation
  - Problèmes de satisfaction des contraintes
  - Exemples des modèles PPC simples
- Méthodes de résolution
  - Recherche arborescente
  - Consistance locale
- Les contraintes globales
- Quelques modèles PPC pratiques
- Solveurs PPC

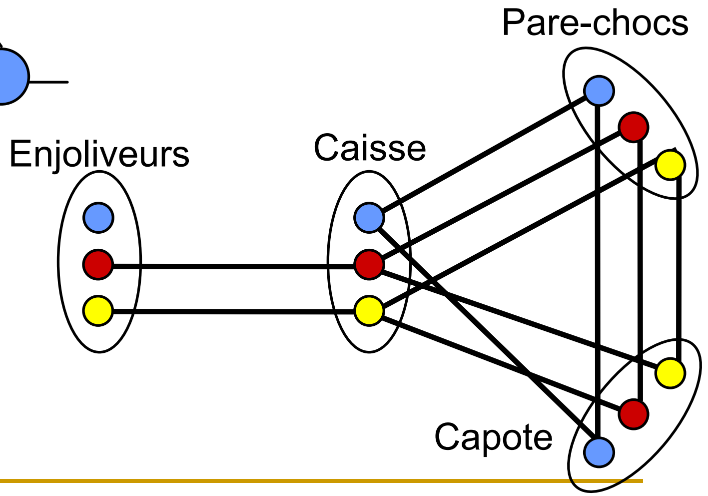
Recherche arborescente

On teste successivement les valeurs possibles.

Si une solution partielle ne vérifie toutes les contraints, on revient à l'étape précédente

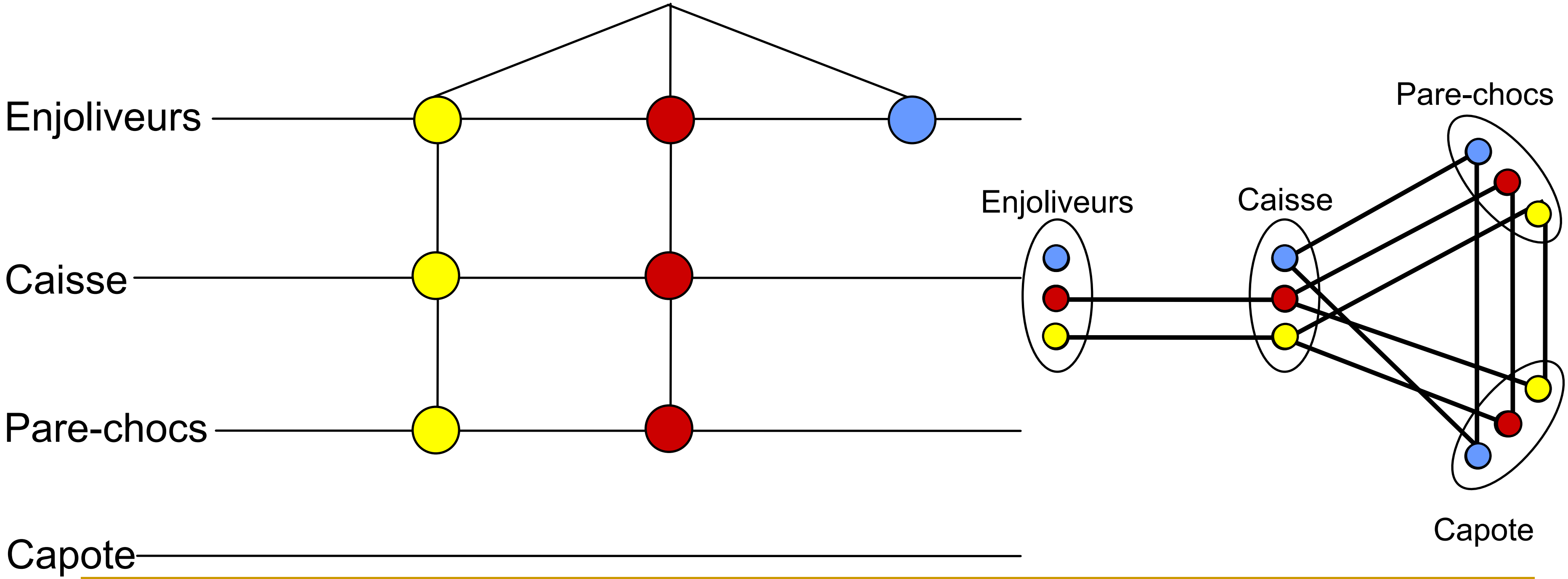
Complexité : O(ed^n), mieux en pratique.



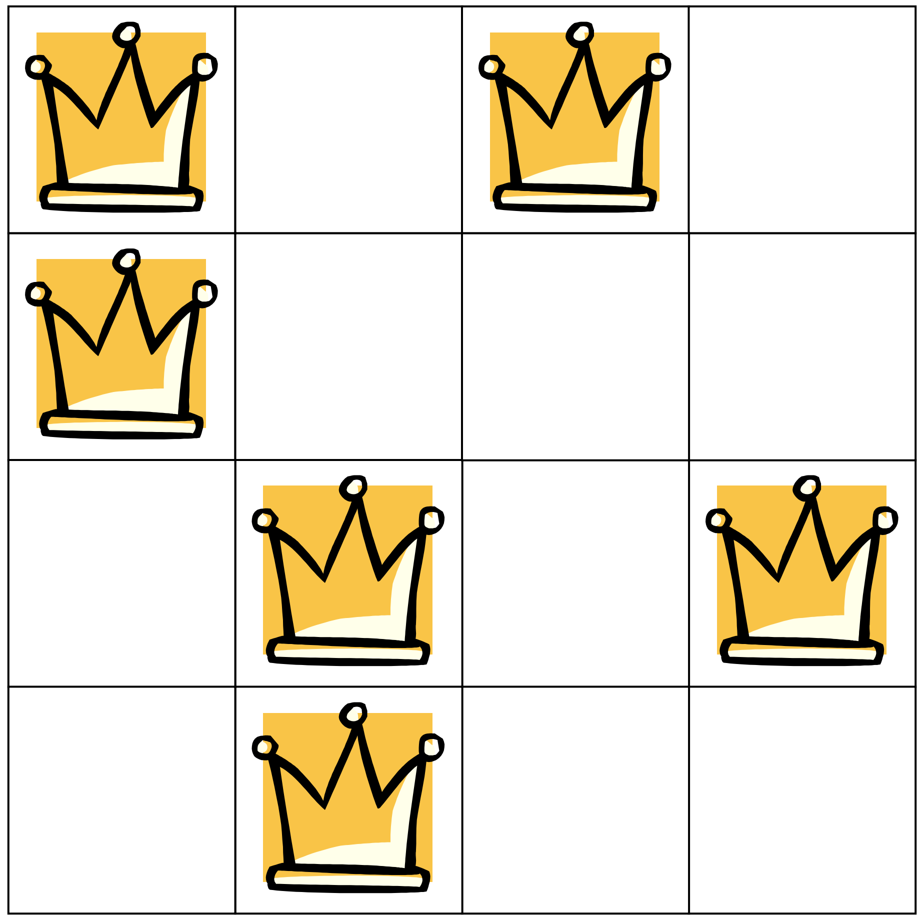


Forward Checking

On choisit une valeur et supprime des domaines toutes les valeurs incompatibles avec la valeur choisie



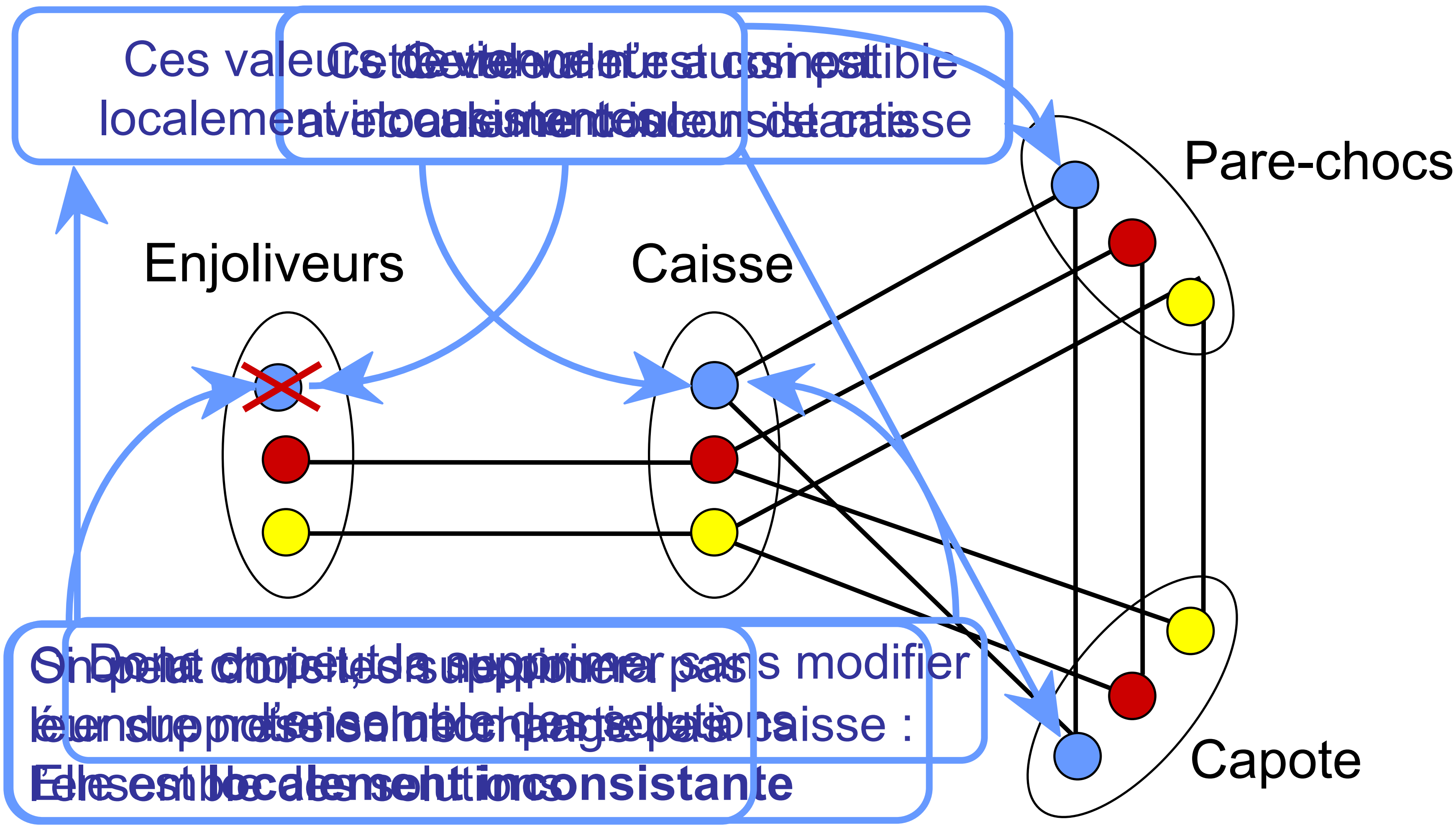
Forward Checking : les N reines



Contenu

- Introduction
- Modélisation
  - Problèmes de satisfaction des contraintes
  - Exemples des modèles PPC simples
- Méthodes de résolution
  - Recherche arborescente
  - Consistance locale
- Les contraintes globales
- Quelques modèles PPC pratiques
- Solveurs PPC

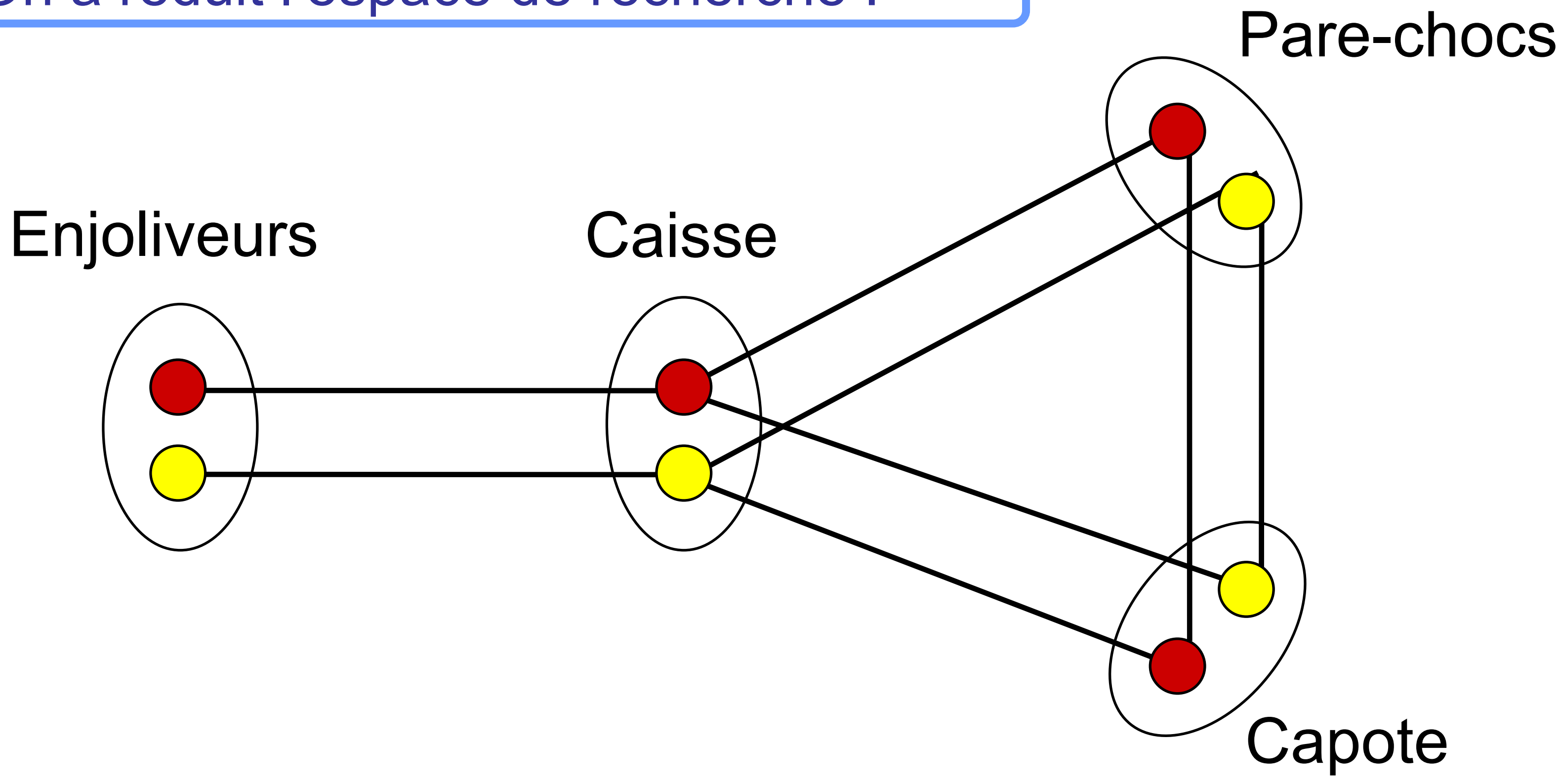
Consistance locale I



Consistance locale I

On n’a pas changé l’ensemble des solutions :
On a un réseau de contraintes **équivalent**

On a réduit l’espace de recherche !



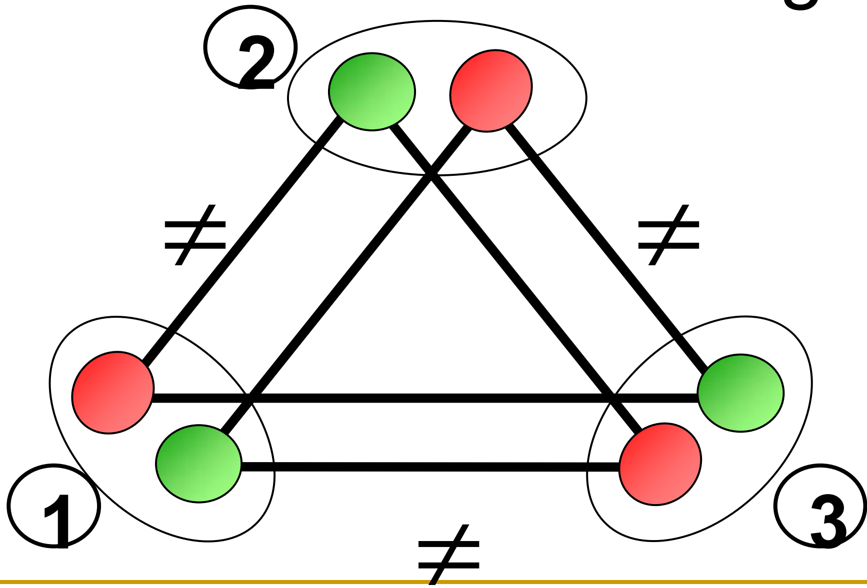
Consistance locale II

- Ce qu'on vient de faire c'est réaliser la fermeture arc-consistante
- La valeur *a* de la variable *i* est arc-consistante ssi elle possède au moins une valeur compatible (un support) dans chaque domaine voisin :

$$\forall C_{ij} \; \exists b \in D_j \text{ tel que } C_{ij}(a,b)$$

Consistance locale III

- Meilleur algorithme pour réaliser la fermeture arc-consistante est AC6 :
  - complexité temporelle est O(ed^2)
  - complexité spatiale est O(ed)
- Arc-consistance $\neq$ Consistance globale

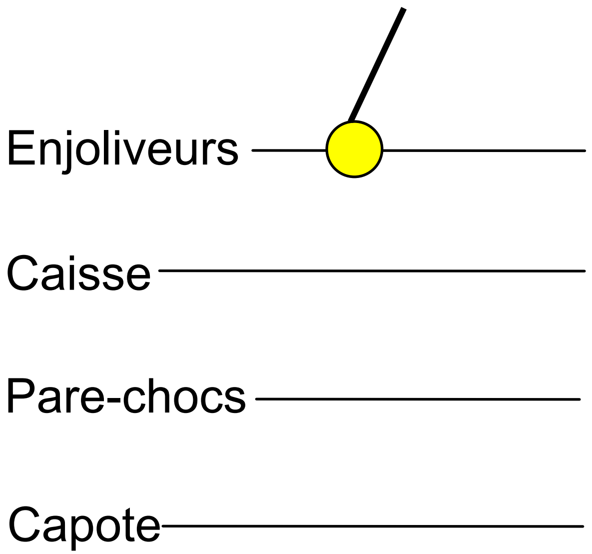


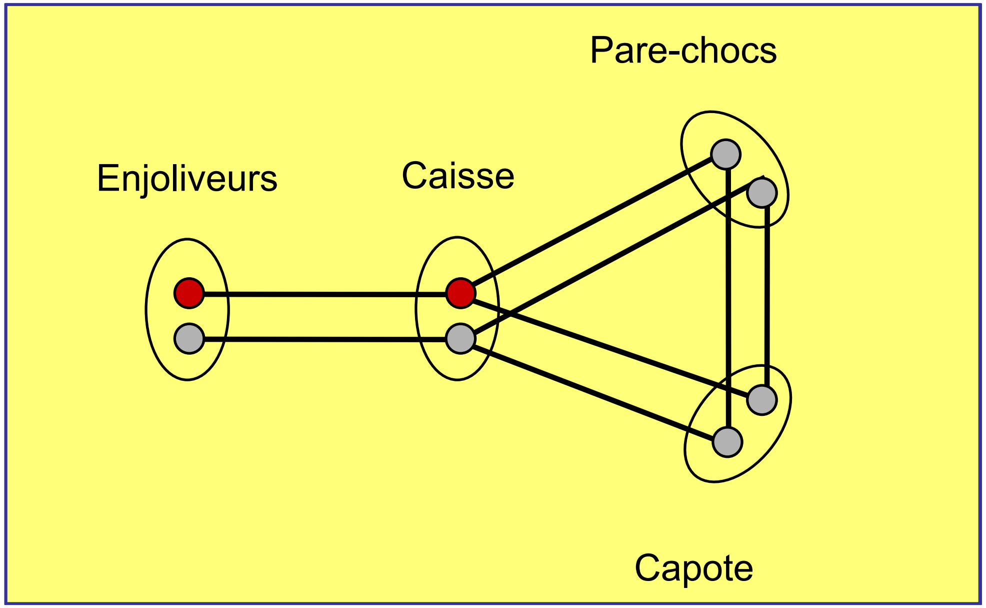
Consistance locale IV : variations

- Arc-consistance
- Path-consistance – extension d'arc-consistance à *k* variables
- Max RPC : path-consistance restreinte
- NI-consistance
- Singleton-consistance
- etc.

Maintenance de l'arc-consistance (MAC)

On choisit une variable et une valeur pour elle, supprime toutes les autres valeurs dans son domaine et toutes les valeurs qui ne sont pas arc-consistantes (propagation)





Options de l'algorithme

- Quelle consistance locale maintenir pendant la recherche?
  - bon rapport coût/puissance
- Quelle variable à instancier d'abord?
  - petit domaine
  - celle qui est impliquée dans de nombreuses contraintes ou les contraintes les plus dures
- Quelle valeur à affecter d'abord?
  - celle qui a le plus de supports

Problèmes de satisfaction des contraintes non-binaires

Comment faire?

- Les transformer en CSP binaires
  - pas très efficace en pratique
- Développer des techniques de propagation spécifiques
  - contraintes arithmétiques
  - contraintes « globales »

Contraintes arithmétiques

- Représentation implicites des contraintes
- Les domaines sont représentés par des intervalles : $D_X = [\min(X),\max(X)]$
- L'arc-B-consistance (arc-consistance restreinte aux bornes de l'intervalle) :

  $C(X_1,\ldots,X_n)$ est arc-B-consistante ssi

  $\forall X_i\ \forall a_i \in \{\min(X_i),\max(X_i)\}\ \exists a_j \in D_j,\ j \neq i$, tq $C(a_1,\ldots,a_n)$

  (plus faible que l'arc-consistance mais facile à implémenter)

Contraintes arithmétiques : exemple ($X=Y+Z$)

- $Z \geq \min(X) + \min(Y)$ et donc
  $\min(Z) = \max\{\min(Z), \min(X) + \min(Y)\}$
  Règle : Quand le minimum de X ou de Y change, recalculer le minimum de Z.
- De même, $Z \leq \max(X) + \max(Y)$

$$X \geq \min(Z) - \max(Y)$$
$$X \leq \max(Z) - \min(Y)$$
$$Y \geq \min(Z) - \max(X)$$
$$Y \leq \max(Z) - \min(X)$$

Contenu

- Introduction
- Modélisation
  - Problèmes de satisfaction des contraintes
  - Exemples des modèles PPC simples
- Méthodes de résolution
  - Recherche arborescente
  - Consistance locale
- Les contraintes globales
- Quelques modèles PPC pratiques
- Solveurs PPC

Contraintes « globales »

Contrainte global est une union des contraintes simples. Les avantages d'utilisation :

- facilitent la modélisation
  - librairies des contraintes
  - moins de contraintes nécessaires
- accélèrent la résolution
  - « on voit plus » si on tient compte de plusieurs contraintes simples au même temps
  - des algorithmes efficaces de propagation

Contrainte « all-different »

all-different($X_1,\ldots,X_n$) – variables $X_1,\ldots,X_n$ doivent prendre des valeurs différents (remplace $n^2/2$ contraintes binaires)

- très souvent utilisée en pratique
- algorithme de propagation polynôme est efficace (**rappel** : algorithme qui élimine des domaines des variables des valeurs localement inconsistantes avec la contrainte)

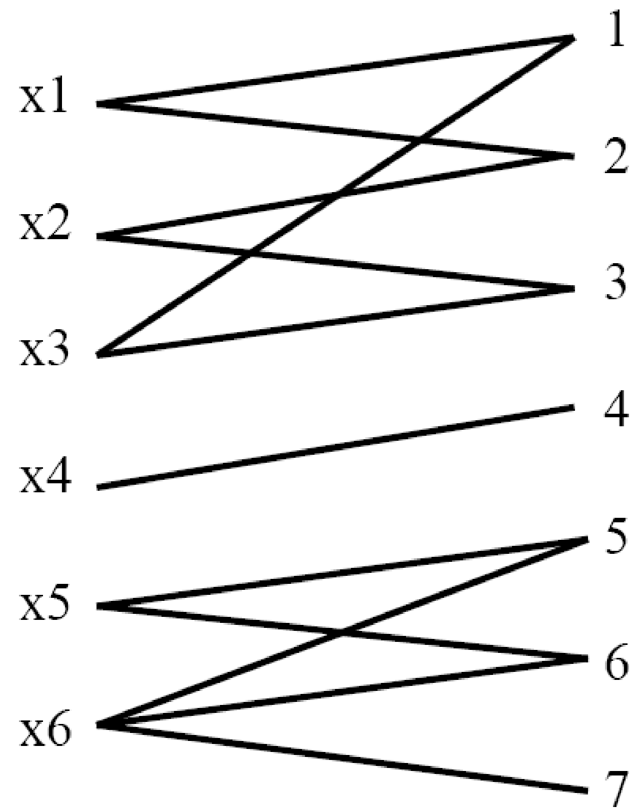
Contrainte « all-different » : propagation

$D_{x1} = \{1,2\}$, $D_{x2} = \{2,3\}$

$D_{x3} = \{1,3\}$, $D_{x4} = \{\cancel{3},4\}$

$D_{x5} = \{\cancel{2},\cancel{4},5,6\}$, $D_{x6} = \{5,6,7\}$

- On trouve un couplage maximum
- On établit les arrêtes qui n'appartiennent pas à aucun chemin ou circuit alternant et qui n'appartiennent pas au couplage
- En éliminant les valeurs correspondantes on obtient l'arc-consistance
- Complexité : *O(nd)*

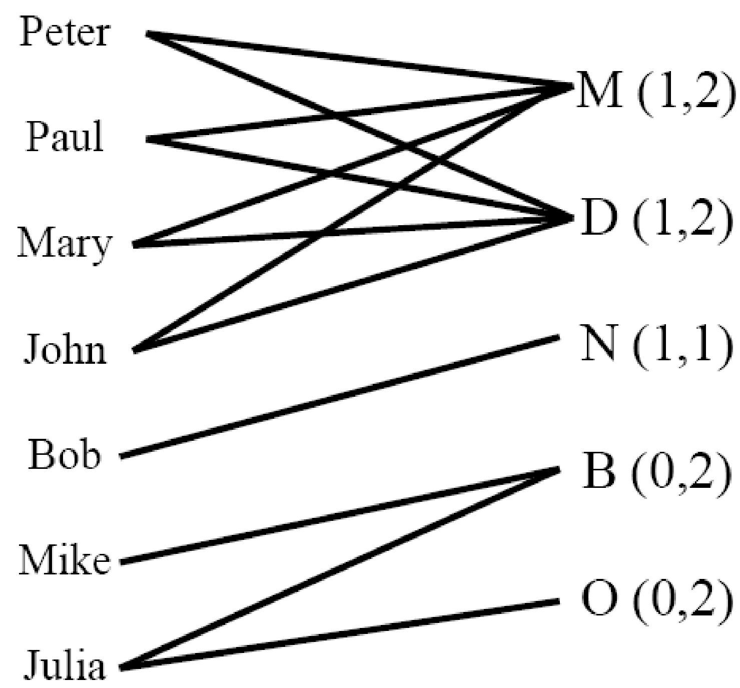


Contrainte GCC

$GCC(X_1,\ldots,X_m,\{v_j,l_j,u_j\}_{j\in N})$ – le nombre de fois chacune valeur v_j est prise par les variables $X_1,\ldots,X_m$ doit être dans l'intervalle $[l_j,u_j]$

M et D doivent être pris par une ou deux personnes, N doit être pris par exactement une personne, B et O doivent être pris par au plus deux personnes

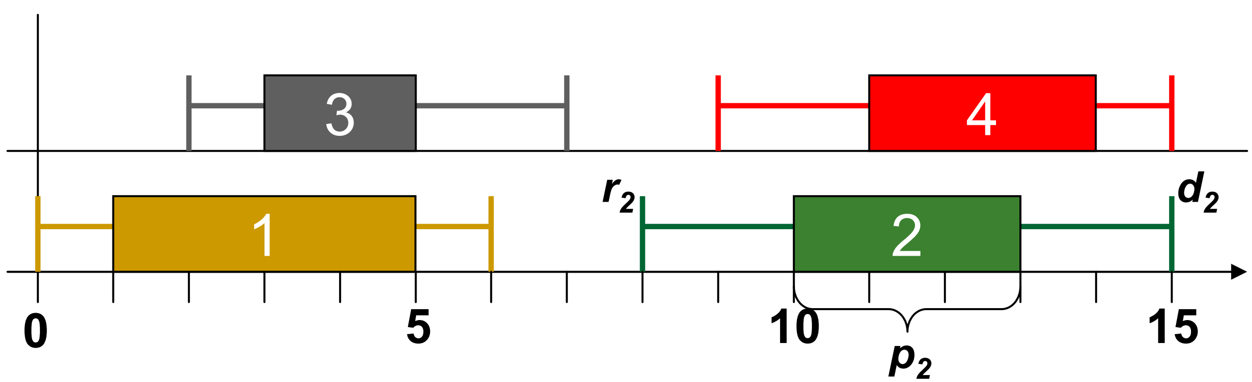
Algorithme de propagation similaire à celui de la contrainte « all-different »



Contrainte « disjunctive »

disjunctive($X_1,\ldots,X_n,p_1,\ldots,p_n$) – remplace $n^2/2$ contraintes binaires : $X_i + p_i \leq X_j$ ou $X_j + p_j \leq X_i$

Équivalent à : n tâches avec $\{r_i,p_i,d_i\}_{i\leq n}$ comme les dates du début, les temps d'exécution, et les dates de fin doivent être exécutées sans chevauchement, domaine de X_i est $[r_i,d_i-p_i]$

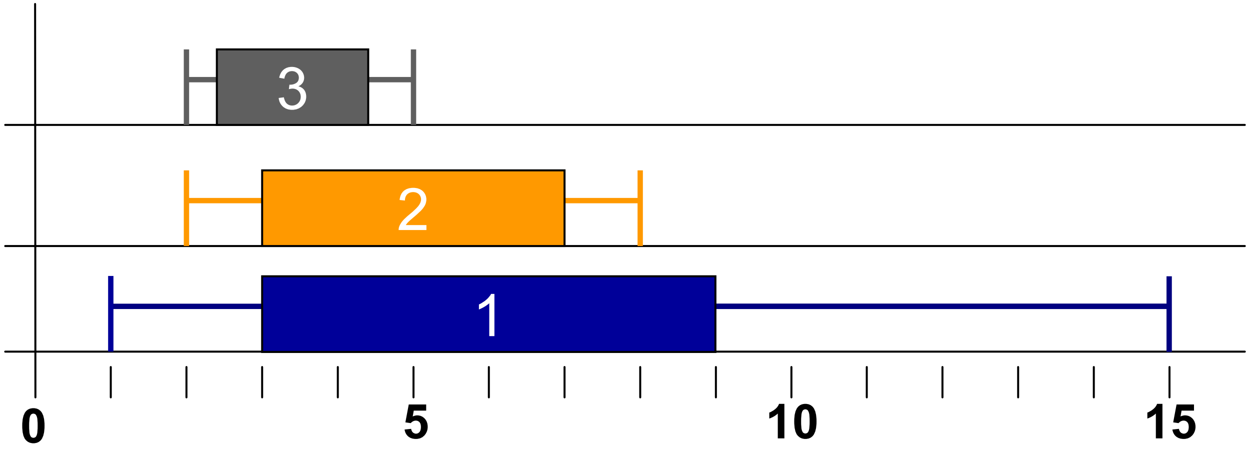


Contraintes « disjunctive » :
la propagation « Edge-Finding »

- Si la tâche *j* ne peut être exécutée que après toutes les tâche d'un ensemble *Ω*, alors

$$X_j \geq \max_{\Omega' \subseteq \Omega}\{r_{\Omega'}+d_{\Omega'}\}$$

- On peut faire toutes les propagations possibles de ce type en temps O(*n*log*n*)



Recherche arborescente : cas non-binaire

Nœud(S, i, a)

$S = S \cup \{(i, a)\}$

supprimer $\{(i, b) \ / \ b \neq a\}$

Faire

pour chaque contrainte C **faire**

propager(C)

Tant que il y a une réduction des domaines

Si il n'existe pas de domaine vide **alors**

Choisir $j \notin S$

pour tout $b \in D_j$ **faire** Nœud(S, j, b)

restaurer les domaines

Contenu

- Introduction
- Modélisation
  - Problèmes de satisfaction des contraintes
  - Exemples des modèles PPC simples
- Méthodes de résolution
  - Recherche arborescente
  - Consistance locale
- Les contraintes globales
- Quelques modèles PPC pratiques
- Solveurs PPC

Modèle : Sudoku

- Variables : X_{ij}, $i,j \in \{1,\ldots,n\times n\}$
- Domaines : $D_{ij} = \{1,\ldots,n\times n\}$
- Contraintes :
  - all-different($X_{k1},\ldots,X_{k,n\times n}$), $k \in \{1,\ldots,n\times n\}$
  - all-different($X_{1k},\ldots,X_{n\times n,k}$), $k \in \{1,\ldots,n\times n\}$
  - all-different($X_{kn+1,\, ln+1},\ldots,X_{kn+n,\, ln+n}$), $k \in \{0,\ldots,n-1\}$, $l \in \{0,\ldots,n-1\}$
- Que des contraintes « all-different » !

$n = 3$

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

Modèle : ordonnancement sportif I

- *n* équipes, *n*-1 semaines, *n*/2 périodes
- chaque paire d'équipes joue exactement 1 fois
- chaque équipe joue un match chaque semaine
- chaque équipe joue au plus deux fois dans la même période

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
Period 1	0 vs 1	0 vs 2	4 vs 7	3 vs 6	3 vs 7	1 vs 5	2 vs 4
Period 2	2 vs 3	1 vs 7	0 vs 3	5 vs 7	1 vs 4	0 vs 6	5 vs 6
Period 3	4 vs 5	3 vs 5	1 vs 6	0 vs 4	2 vs 6	2 vs 7	0 vs 7
Period 4	6 vs 7	4 vs 6	2 vs 5	1 vs 2	0 vs 5	3 vs 4	1 vs 3

source : *Jean-Charles Régin*

Modèle : ordonnancement sportif II

- Pour chaque case, il y a 2 variables qui représentent les équipes : *Tpwh* et *Tpwa*, $p \in [1,\ldots,n/2]$, $w \in [1,\ldots,n\text{-}1]$, $D_{Tpwh}=\{1,\ldots,n\}$, $D_{Tpwa}=\{1,\ldots,n\}$, $Tpwh<Tpwa$
- Pour chaque case, il y a une variable représentant le match : *Mpw*, $p \in [1,\ldots,n/2]$, $w \in [1,\ldots,n\text{-}1]$, $D_{Mpw} = \{1,\ldots,n(n\text{-}1)/2\}$, $Mpw = n \times Tpwh+Tpwa$.

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
Period 1	M11	M12	M13	M14	M15	M16	M17
Period 2	M21	M22	M23	M24	M25	M26	M27
Period 3	M31	M32	M33	M34	M35	M36	M37
Period 4	M41	M42	M43	M44	M45	M46	M47

Modèle : ordonnancement sportif III

- all-different($\{Mpw\}_{p \le n/2, w \le n-1}$)
- all-different($\{Tpwh, Tpwa\}_{p \le n/2}$), $w \in [1,\ldots,n-1]$
- gcc($\{Tpwh, Tpwa\}_{w \le n-1}$, $\{k,1,2\}_{k \le n-1}$), $p \in [1,\ldots,n/2]$
- symétrie (très important!)

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Dummy
Period 1	0 vs 1	0 vs 2	4 vs 7	3 vs 6	3 vs 7	1 vs 5	2 vs 4	5 vs 6
Period 2	2 vs 3	1 vs 7	0 vs 3	5 vs 7	1 vs 4	0 vs 6	5 vs 6	2 vs 4
Period 3	4 vs 5	3 vs 5	1 vs 6	0 vs 4	2 vs 6	2 vs 7	0 vs 7	1 vs 3
Period 4	6 vs 7	4 vs 6	2 vs 5	1 vs 2	0 vs 5	3 vs 4	1 vs 3	0 vs 7

Modèle : ordonnancement sportif IV

- En utilisant PPC, on peut trouver un ordonnancement pour 40 équipes (dans 6h)
- Taille réelle!
- Aujourd'hui, les ordonnancements pour MLB (centaines des contraintes) sont produits en utilisant la Recherche Opérationnelle (PLNE, PPC, heuristiques)

source : *Michael A. Trick*

Modèle : emplois du temps I

- Il y a 3 équipes : jour (D), soir (E), nuit (N)
- Variables : Jik, $D_{Jik} = \{D,E,N,–\}$
- Variables : Lik, $D_{Lik} = \{1.0,0.8,0.5,0.0\}$
- Contrainte : $Jik=D \Rightarrow Lik=1.0$, …
- Contrainte : $gcc(\{Jik\}_{\forall i},\{D_{Jik},0,1\}), \forall k$
- Contrainte : $\Sigma_{\forall k}\, Lik \geq 3.0, \forall i$

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
M. Green	D	-	D	-	D	-	D
Mrs. Blue	-	N	N	N	N	N	N
M. Red	N	D	-	D	E	D	-
M. Yellow	E	E	E	E	-	E	E

source : ***Jean-Charles Régin***

Modèle : emplois du temps II

- Longueur de série : stretch($\{Jik\}_{\forall k}$,{2,2,2,2},{4,4,4,4}), $\forall$i,
- Pas de changement d'équipe sans un repos, rotation en avant (D… E… N… D) regular($\{Jik\}_{\forall k}$,MOTIF)
- Préférences individuelles, congés, formations,…

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
M. Green	D	D	-	E	E	E	E
Mrs. Blue	E	E	E	-	N	N	N
M. Red	N	N	N	N	-	D	D
M. Yellow	-	-	D	D	D	-	-

Emplois du temps, instances réelles

°	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S
23796	-	-	-	-	D	D	N	N	-	-	D	-	-	-	-	D	-	D	D	-	D	D	-	-	-	-	-	-
603042	D	D	D	D	E	-	-	-	D	D	D	D	-	D	D	D	D	D	E	-	-	-	D	D	D	D	-	D
12310	D	D	-	-	-	-	-	-	-	-	-	-	-	D	D	D	-	-	-	-	-	-	-	-	-	-	-	D
511811	D	D	D	-	D	D	-	-	D	D	-	-	-	D	D	D	D	-	D	D	-	-	D	D	-	-	-	D
60324	-	-	D	D	D	-	D	D	-	D	D	D	-	-	-	-	-	D	D	-	D	D	D	-	D	D	D	-
603095	E	-	-	E	E	E	-	-	-	-	-	-	E	E	E	-	-	E	E	E	-	-	-	-	E	-	-	E
603230	-	D	D	D	D	-	D	D	D	D	-	D	D	-	-	D	D	D	D	-	D	D	D	-	D	D	D	-
510723	D	D	D	-	-	D	-	-	D	D	D	-	-	D	D	D	D	-	-	D	-	-	D	D	D	-	-	D
511104	-	R	R	R	R	R	-	-	R	R	R	R	R	-	-	-	-	E	E	-	E	E	-	-	E	E	E	-
34108	-	D	D	D	D	-	D	D	D	D	-	-	-	-	-	R	R	R	R	R	D	D	-	-	D	-	-	-
11866	-	D	-	D	D	D	E	E	-	-	D	-	-	-	-	D	-	D	D	D	E	E	-	D	-	-	-	-
35022	-	R	R	R	R	R	D	D	-	-	-	-	-	-	-	-	-	-	D	-	D	D	D	-	D	D	D	-
512287	E	E	E	-	D	D	E	E	-	-	-	-	-	E	E	E	E	-	D	-	E	E	-	-	E	-	-	E
56507	D	D	-	D	D	D	-	-	D	-	-	-	-	D	D	D	-	D	D	D	-	-	D	-	-	-	-	D
512281	-	E	-	D	D	-	D	D	E	-	-	-	-	-	-	E	-	D	D	-	D	D	E	-	-	-	-	-
511066	-	D	D	-	-	-	D	D	-	-	-	-	D	-	-	-	-	-	-	-	D	D	-	-	D	D	D	-
600955	D	D	-	D	D	-	-	-	-	-	-	-	-	D	D	D	-	D	D	-	-	-	-	-	-	-	-	D
602576	D	D	-	D	D	D	-	-	-	-	-	-	-	D	D	D	-	D	D	D	-	-	-	-	-	-	-	D
600315	-	-	T	T	-	-	T	T	-	T	-	T	T	-	-	-	T	-	-	T	T	T	-	-	T	T	T	-
511865	-	-	-	-	-	-	T	T	-	T	T	T	T	-	-	-	-	-	-	-	-	-	R	R	R	R	R	T

Pour quels problèmes il faut choisir la PPC?

- Contraintes complexes
- Grand nombre de contraintes (pas beaucoup de solutions réalisables)
- Problèmes de décision
- Problèmes avec fonction objective « simple »
- Problèmes issu des domaines où PPC était réussi

Contenu

- Introduction
- Modélisation
  - Problèmes de satisfaction des contraintes
  - Exemples des modèles PPC simples
- Méthodes de résolution
  - Recherche arborescente
  - Consistance locale
- Les contraintes globales
- Quelques modèles PPC pratiques
- Solveurs PPC

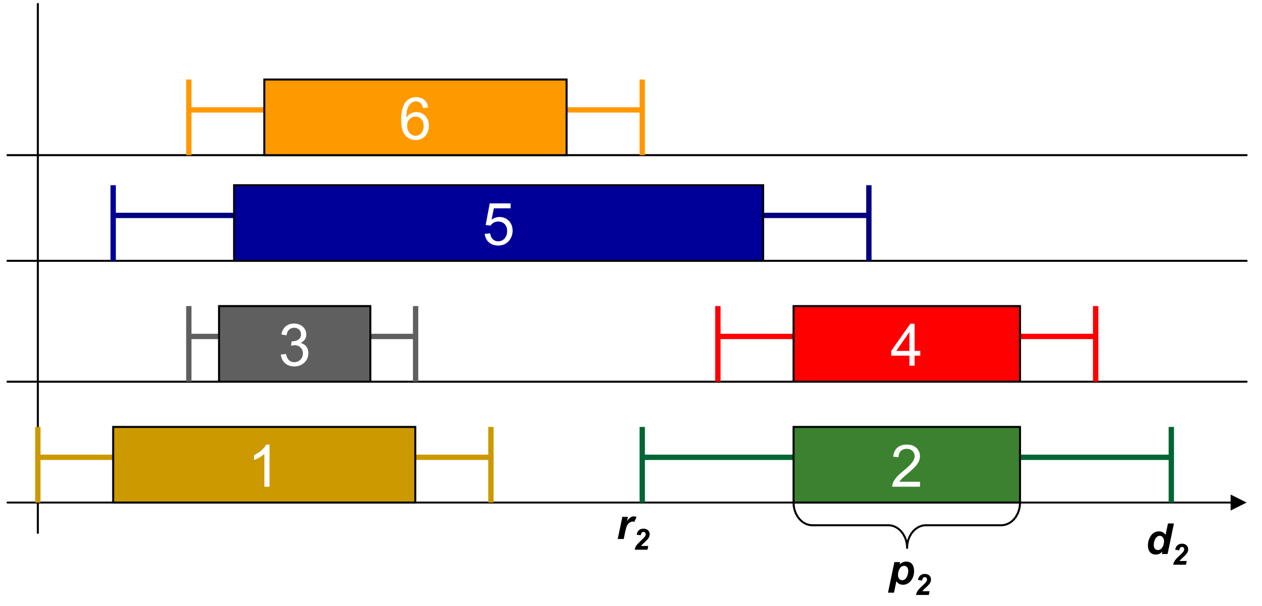
Solveurs PPC

- Gratuites
  - Choco (http://choco.sourceforge.net/)
  - Gecode (http://www.gecode.org)
  - …
- Payants
  - ILOG CP (http://www.ilog.com/products/cp/)
  - Xpress Kalis (http://www.dashoptimization.com)
  - …

TD : affectation des tâches sur des machines hétérogènes (ATMH)

- n tâches, m machines
- chaque tâche *j* a une date de disponibilité r_j, un deadline d_j, des temps de procession p_{ij}, et des coûts de procession c_{ij}.
- chaque machine peut exécuter au plus une tâche à la fois, les interruptions sont interdites
- il faut trouver une affectation réalisable des tâches sur les machines de coût minimum

ATMH : une solution réalisable



1 4 5		M1
2 3 6		M2

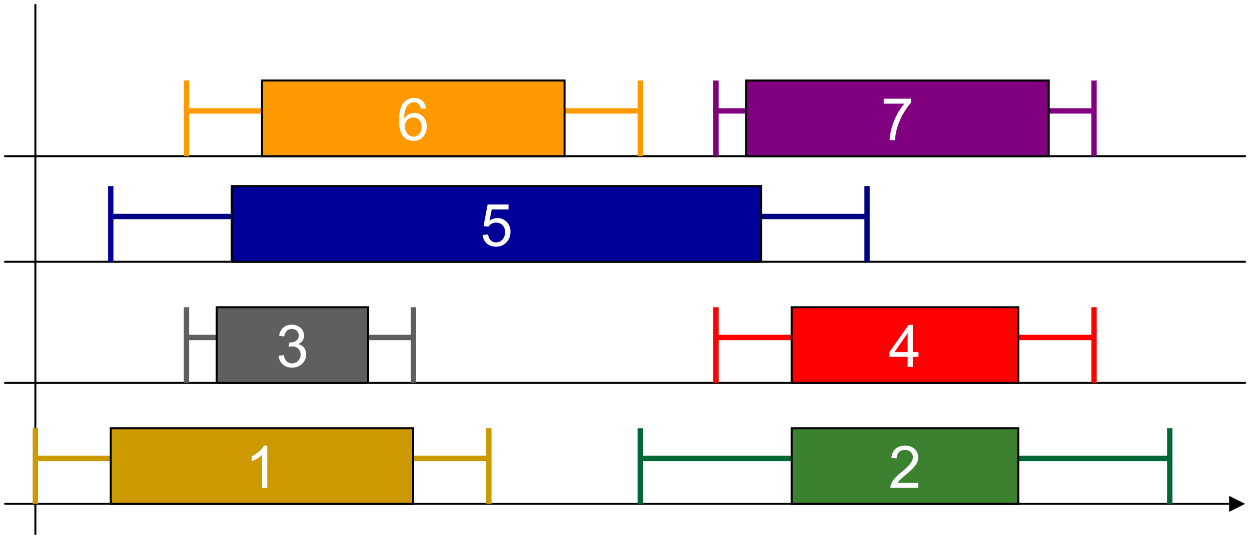
$$\text{Coût : } c_{11} + c_{14} + c_{15} + c_{22} + c_{23} + c_{26}$$

ATMH : une formulation PLNE

X_{ij} = 1 ssi la tâche *j* est affectée sur la machine *i*

$$
\begin{aligned}
\min \quad & \sum_{i\in M}\sum_{j\in N} c_{ij}X_{ij} \\
& \sum_{i\in M} X_{ij} = 1, \quad \forall j \in N, \\
& \sum_{j\in N} p_{ij}X_{ij} \le \max_{j\in N} d_j - \min_{j\in N} r_j \quad \forall i \in M, \\
& \sum_{S\subseteq N} X_{ij} \le |S| - 1, \quad \forall i \in M,\ S \text{ est non-réalisable sur } i \\
& X_{ij} \in \{0,1\}, \quad \forall i \in M, \forall j \in N.
\end{aligned}
$$

ATMH : génération des contraintes



1 6 2 7		M1

Contrainte additionelle:

$$X_{11} + X_{12} + X_{16} + X_{17} \leq 3$$

ATMH : l'algorithme

- On élimine toutes les contraintes de réalisabilité
- **Faire**
  - On résout le PLNE
  - **Pour chaque** machine $i \in M$ **faire**
    - S est un sous-ensemble des tâches affectées sur i par PLNE
    - On vérifie si il y a un ordonnancement réalisable pour S sur i (PPC : une seule contrainte disjunctive($\{r_j\}_{j \in S}$,$\{p_{ij}\}_{j \in S}$,$\{d_j\}_{j \in S}$))
    - **Si** il n'y en a pas **alors**
      - on ajoute la contrainte $\sum_{j \in S} X_{ij} \leq |S| - 1$ au PLNE
    - **fin-si**
  - **fin-pour**
- **Tant que** il y a des contraintes ajoutées