Intelligence Artificielle
Satisfaction de contraintes

PLAN

- Problèmes à satisfaction de contraintes (PSC) - définition
- PSC comme problème de recherche
- Algorithme de "backtracking"
- Heuristiques générales
- Technique de recherche locale

Définition

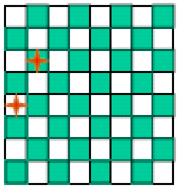
- Un problème à satisfaction de contraintes (PSC) est constitué:
  - d'un ensemble de variables {X1, X2, … , Xn}
    chaque variable Xi ayant un domaine Di de valeurs possibles
  - habituellement Di est discret et fini
  - d'un ensemble de contraintes {C1, C2, … , Cp}

- chaque contrainte Ck concerne un sous-ensemble de variables et spécifie les combinaisons autorisées pour les valeurs de ces variables

- But: assigner une valeur à chaque variables de sorte que toutes les contraintes soient satisfaites.

- Exemples: 8-reines, arithmétique cryptée, coloration de cartes, disposition des éléments sur un circuit VLSI, ordonnancement, ...

Exemple : problème des 8-reines

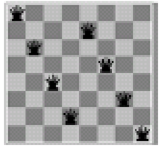


-- 64^8 possibilités avec test/erreur.

-- PSC : 1ère définition:

- **64 variables Xij, i = 1 à 8 et j = 1 à 8**
- **Domaine de chaque variable**
{ 1, 0 }
- **Contraintes de la forme:**
– Xij = 1 ; Xik = 0 pour tout k = 1 à 8, k ≠ j
– Xij = 1 ; Xkj = 0 pour tout k = 1 à 8, k ≠ i
– Contraintes semblables pour les diagonales
– ΣXij = 8

-- PSC : 2ème définition:



Simplification
- **8 variables Xi, i = 1 à 8**
- **Domaine de chaque variable:**
{1, 2, 3, 4, 5, 6, 7, 8}
- **Contraintes de la forme:**
– Xi = k , Xj ≠ k pour tout j = 1 à 8, j ≠ i
– Idem pour les diagonales

Exemple: coloration de carte

- **7 variables: {WA, NT, SA, Q, NSW, V, T}**
- **Chaque variable a le même domaine {rouge, vert, bleu}**
- **Contraintes: 2 régions adjacentes doivent avoir des couleurs différentes**

– WA≠NT, WA≠SA, NT≠SA, NT≠Q, SA≠Q, SA≠NSW, SA≠V,Q≠NSW, NSW≠V

– ou: (WA, NT) dans **{(rouge,vert), (rouge,bleu), (vert,rouge), (vert,bleu) …}**

- **Les solutions sont des affectations satisfaisant toutes les contraintes, exemple:**

{WA=rouge, NT=vert, Q=rouge, NSW=vert, V=rouge, SA=bleu, T=vert}

Exemples de PSC réels

- Problèmes d'affectation
  - ex: qui enseigne quel cours?
  - Aménagement d’espace
- Configuration matérielle
- Organisation de transport (chemins de fer, compagnies aériennes)
- Ordonnancement de production (atelier)
- ---------------------------------

Beaucoup de problèmes du monde réel impliquent des variables à valeurs réelles (continues)

PSC comme problème de recherche

- État initial: Affectation vide { }
- Fonction successeur: une valeur est assignée à chaque variable libre, sans que cela n'entre en conflit avec les variables ayant déjà reçu une valeur
- Test-solution: l'affectation est complète
- Si on a n variables avec des domaines de taille d $\Rightarrow$ $O(d^n)$ affectations complètes distinctes
- Ceci est valable pour tous les PSC !
- Si on a n variables, toute solution apparaîtra à une profondeur n $\Rightarrow$ on peut utiliser une recherche en profondeur
- La notion de chemin est non pertinente !

Commutativité des PSC

• L'ordre dans lequel les valeurs sont assignées aux variables est sans importance pour la solution finale

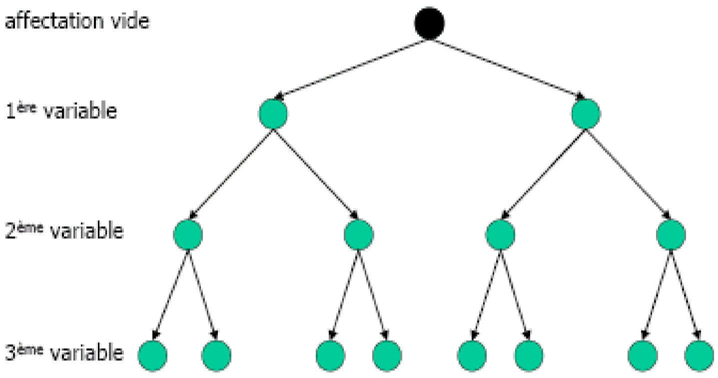
– [WA=rouge suivi de NT=vert] identique à [NT=vert suivi de WA=rouge]

• Donc:
– Prolonger un noeud en ne considérant l'affectation que d'une seule variable à la fois
– Ne pas mémoriser le chemin menant à un noeud donné

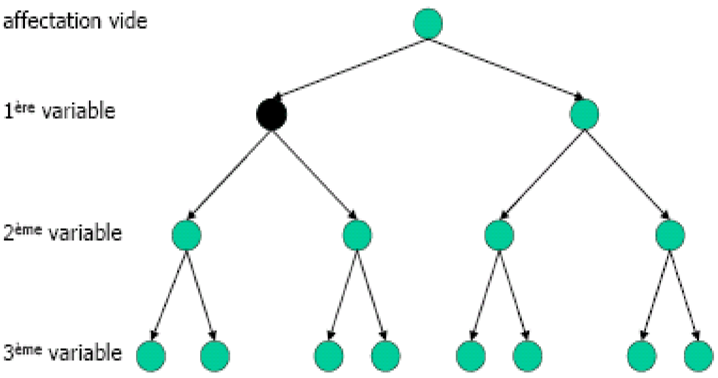
• La recherche en profondeur pour des PSC avec affectation d'une seule variable à la fois est appelée recherche "backtracking"

– c'est l'algorithme non heuristique de base pour les PSC
– capable de résoudre le problème des n-reines pour $n \approx 25$

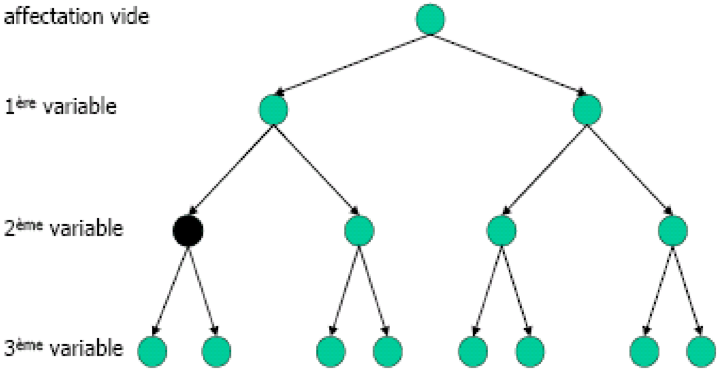
Recherche "backtracking"



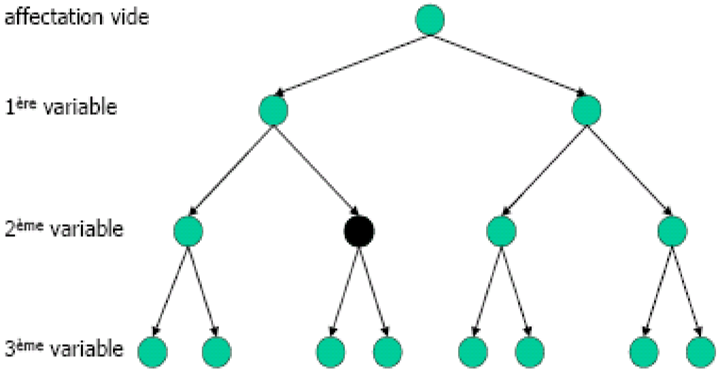
Affectation = { }



Affectation = {(var1=v11)}



Affectation = {(var1=v11),(var2=v21)}



Affectation = {(var1=v11),(var2=v22)}

Algorithme de "backtracking"

PSC-BACKTRACKING({ }) // l’appel de départ

PSC-BACKTRACKING(**a**) // **a**: Affectation partielle des variables
Si **a** est complet alors retourner **a**
Sinon
– X□ sélectionner une variable non assignée

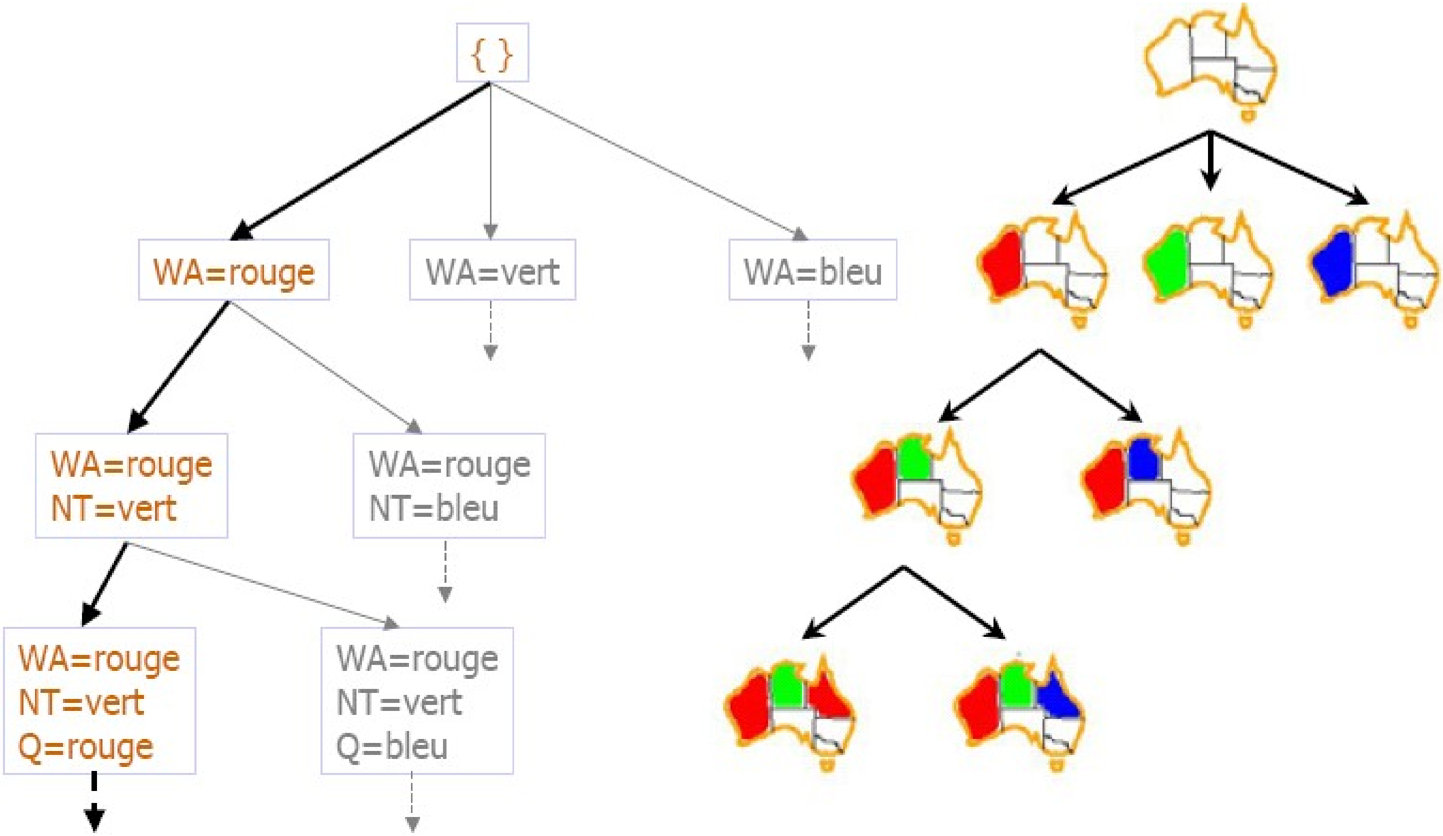
– D□ définir un ordre pour le domaine D de X

– Pour chaque valeur v de D faire

- Si v est consistante avec **a** alors
  – Ajouter (X= v) à **a**
  – résultat □ PSC-BACKTRACKING(**a**)
  – Si résultat ≠ *échec* alors retourner **résultat**

– Retourner *échec*

Exemple: coloration de carte



Questions

- Quelle variable X doit être prise en considération à la prochaine étape?
- Dans quel ordre faut-il trier les valeurs de son domaine ?
- Peut-on détecter un échec inévitable assez tôt ?
- Peut-on tirer avantage de la structure du problème ?
- Quelles sont les conséquences d'une affectation partielle (solution partielle) sur les variables pas encore assignées ?

(→ Problème de la propagation de contraintes)

Choix de la variable

- Heuristique de la variable la plus contrainte
  - sélectionner la variable avec le plus petit nombre de valeurs possibles
- Heuristique de la variable la plus contraignante
  - sélectionner la variable qui est impliquée dans le plus grand nombre de contraintes sur les variables pas encore assignées

Choix de la valeur

- Heuristique de la valeur la moins contraignante
  - préférer la valeur qui laisse le plus de valeurs possibles pour les autres variables pas encore assignées
- Une combinaison de ces différentes heuristiques rend le problème des 1000-reines praticable

Remarque

• La recherche locale avec heuristique de minimisation de conflit est efficace pour le problème des n-reines avec n = plusieurs millions

• La raison en est que:

les solutions étant densément distribuées dans l'espace de dimension $O(n^n)$, on n'est, en moyenne, jamais qu'à un petit nombre d'étapes d'une solution à partir d'une affectation choisie au hasard

Procédure (recherche locale):

• Choisir au hasard une affectation complète

• Répéter

  – Choisir au hasard une variable x qui est en conflit avec une contrainte

  – Fixer une nouvelle valeur pour x qui minimise le nombre de conflits

  – Si la nouvelle affectation est sans conflit alors c'est une solution

(heuristique de la minimisation des conflits)

PSC à domaine infini

• Ce sont les problèmes pour lesquels les domaines des variables sont l'ensemble des nombres entiers (PSC discrets) ou celui des nombres réels (PSC continus)

• Les contraintes sont alors exprimées par des égalités ou par des inégalités

• Cas particulier: les problèmes de programmation linéaire

Applications

- Les techniques de PSC permettent de résoudre des problèmes très complexes
- De nombreuses applications telles que:

– affectation d'équipages à des lignes aériennes
– gestion d'une flotte de transport
– horaires de trains, d'avions, etc …
– ordonnancement et gestion des tâches dans un port marchand
– conception (en tous genres)
– opérations chirurgicales (neurochirurgie)
– etc…

Références

- Ouvrages
  - Marriott and Stuckey, 1998
  - AIMA, Russell and Norvig, 2nd ed.

- Internet
  - Constraints Archive
http://www.cs.unh.edu/ccc/archive