

---

# Chapitre 01

## Partie 4

---

RESPONSABLE MODULE

DR. HADJADJ ABDELHALIM



Centre universitaire de Mila

Master I: Matière Intelligence artificielle : Principes et Applications

Centre universitaire de Mila

Master I: Matière Intelligence artificielle : Principes et Applications

# Introduction aux Problèmes à Satisfaction de Contraintes

---

Résoudre certains problèmes:

- ✓ Les formulant comme problème de recherche dans un graphe d'états
- ✓ Nœud est une configuration (état) de l'environnement  
Fonction de transition reflète les propriétés de l'environnement  
Heuristique ( $h$ ) pour guider efficacement l'exploration.

- l'algorithme de recherche ne sait pas comment choisir des successeurs d'un nœud est fait par la fonction de transition
- Les nœuds du graphe sont « obscur » vis-à-vis de l'algorithme de recherche.

# Introduction aux Problèmes à Satisfaction de Contraintes

---

CSP constituent cas particulier de la recherche heuristique

- Nœud= ensemble de variables avec des valeurs correspondantes
- Transitions entre les nœuds tiennent compte de contraintes sur les valeurs possibles des variables
- Traduisant un problème sous forme de satisfaction de contraintes
- On élimine la difficulté de définir l'heuristique  $h(n)$  pour notre application

# Exemples

## Sudoku

- Variables: Chaque case de la grille est une variable, à laquelle on doit assigner une valeur de 1 à 9.

- Contraintes :

- - Chaque ligne doit contenir les chiffres de 1 à 9 sans répétition.
- - Chaque colonne doit également contenir les chiffres de 1 à 9 sans répétition.
- - Chaque sous-grille (3x3) doit contenir les chiffres de 1 à 9 sans répétition.

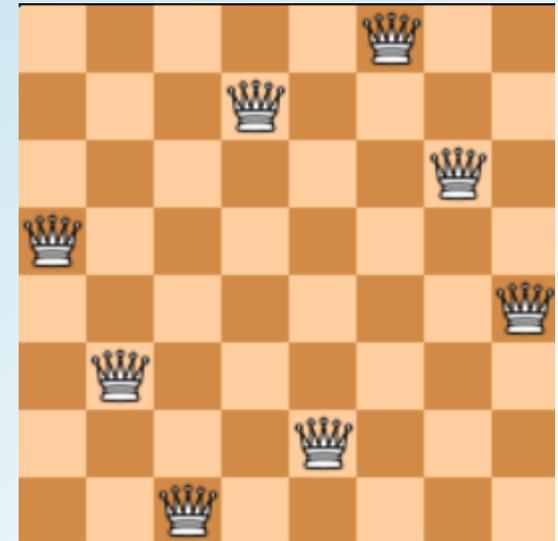
- Objectif : Assigner les valeurs aux cases de manière à satisfaire toutes les contraintes.

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

# Exemples

---

Placer  $n$  reines sur un échiquier  $n \times n$  tel que deux reines ne soient pas en prises



# Exemples

---

- Variables : Les cours à planifier, les locaux disponibles, les créneaux horaires, les professeurs et les étudiants.
- Contraintes:
  - - Un cours donné doit avoir un local assigné à un moment spécifique.
  - - Un local ne peut accueillir qu'un seul cours à la fois.
  - - Un étudiant ou un professeur ne peut pas avoir deux cours en même temps.
  - - Les cours doivent respecter les préférences horaires des professeurs ou des étudiants (si applicable).
- Objectif: Assigner chaque cours à un créneau et un local, en s'assurant que toutes les contraintes sont respectées.

# Définition des Problèmes Satisfaction des Contraintes

---

CSP est ensemble des problèmes, définis par des contraintes, et constituant a chercher une solution les respectant.

# Composantes d'un CSP

---

- **Variables** : Les éléments que l'on doit assigner.
- **Domaines** : Les ensembles de valeurs possibles pour chaque variable.
- **Contraintes** : Les règles qui limitent les valeurs que les variables peuvent prendre.

# résolution de ces CSPs

---

- ❑ Un état d'un problème CSP est défini par une assignation de valeurs à certaines variables ou à toutes les variables.  $\{X_i=v_i, X_n=v_1, \dots\}$
- ❑ Assignation Compatible -----Viole aucune contrainte
- ❑ Assignation Complète ----- Concerne toutes les variables
- ❑ Solution de CSP ----- Compatible + Complète

# Contrainte

---

- ❑ Une contrainte est une relation logique établie entre différentes variables, chacune prenant sa valeur dans un ensemble qui lui est propre, appelé domaine.
- ❑ Une contrainte est déclarative et relationnelle puisqu'elle définit une relation entre les variables sans spécifier de procédure opérationnelle pour assurer cette relation.
- ❑ *L'ordre dans lequel sont posées les contraintes n'est pas significatif* : la seule chose importante à la fin est que toutes les contraintes soient satisfaites

# Représentation des CSP

---

## Définition formelle

Un CSP  $P=(X, D, C)$  est défini par:

- Des Variables:  $\{X_1, \dots, X_n\}$  ----- Ensemble fini de variables
- Des domaines:  $D=\{D_1, D_2, \dots, D_n\}$  (Chaque variable  $X$  a un domaine  $D$  de valeurs possible)
- Des Contrainte:  $\{C_1, \dots, C_M\}$  ---- Ensemble fini de contraintes (une contrainte  $C_i$  est une relation défini sur un ensemble de variables)

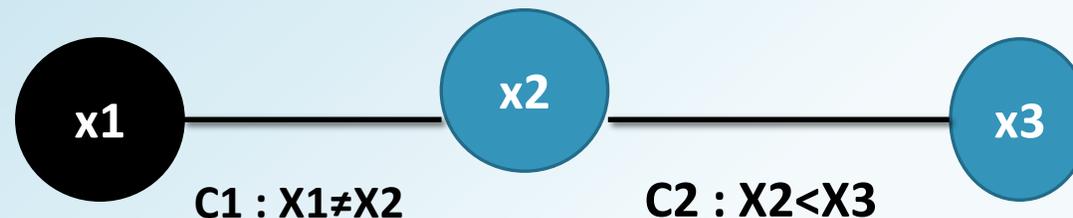
# Représentation des CSP: Définition formelle

## Graphe de Contraintes :

- Un graphe de contraintes est une représentation visuelle d'un problème à satisfaction de contraintes, où :
  - **Nœuds** : Représentent les variables du CSP.
  - **Arêtes** : Représentent les contraintes qui existent entre les variables.

exemple:

- $D(X1) = \{1, 2\}$
- $D(X2) = \{1, 2, 3\}$
- $D(X3) = \{2, 3\}$



# Modélisation d'un CSP

---

- ▣ Identifier les variables : les inconnues
  - ▣ Identifier les domaines de valeur de ces variables
  - ▣ Identifier les contraintes
- 
- Un même problème peut généralement être modélisé par différents CSP.
    - Efficacité : taille de l'espace de recherche de solutions

# Algorithmes de résolution d'un CSP

---

- génère et teste (GET)
- simple retour arrière (SRA) ou (backtracking (BT))
- consistance de nœud (NC), d'arc (AC), de chemin (PC) . . .
- ✓ forward checking (FC)
- ✓ look ahead (LH)
- heuristiques

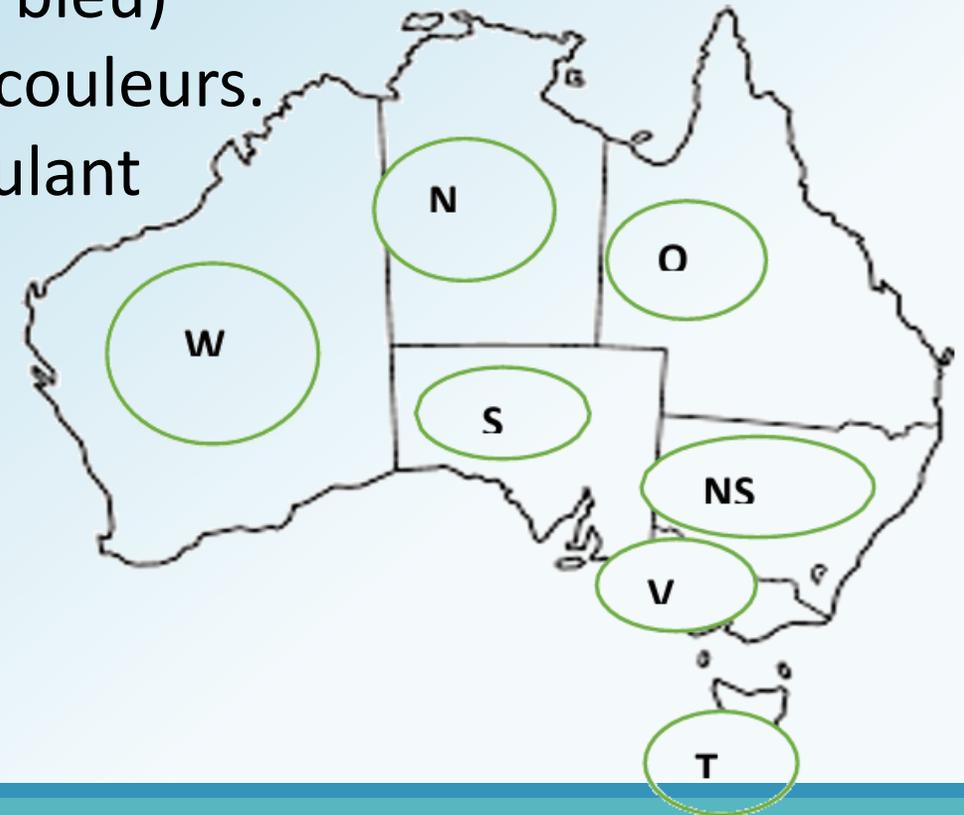
# Colorage d'une carte

- Une carte de l'Australie
- Utiliser seulement trois couleurs (rouge, vert et bleu)
- Deux états frontaliers n'aient jamais les mêmes couleurs.
- Trouver une solution à ce problème en le formulant comme un problème CSP.

$V = \{ \}$

Domaine des variables =  $\{ \}$

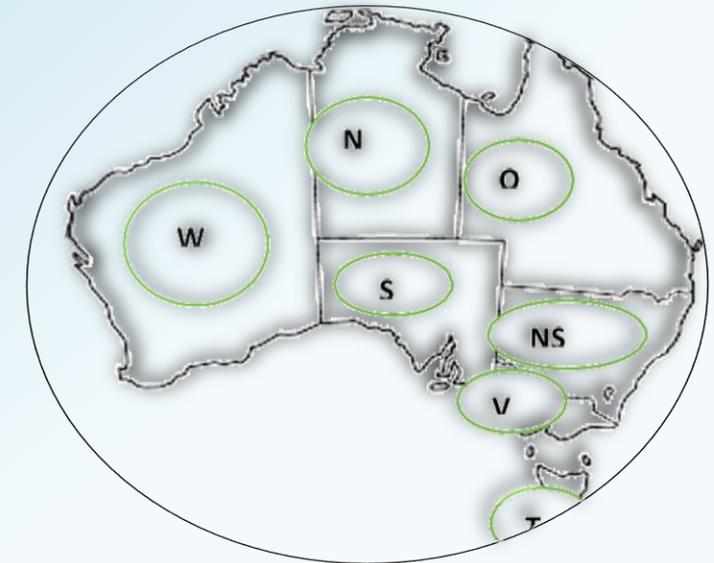
Contraintes =  $\{ \}$



# Colorage d'une carte

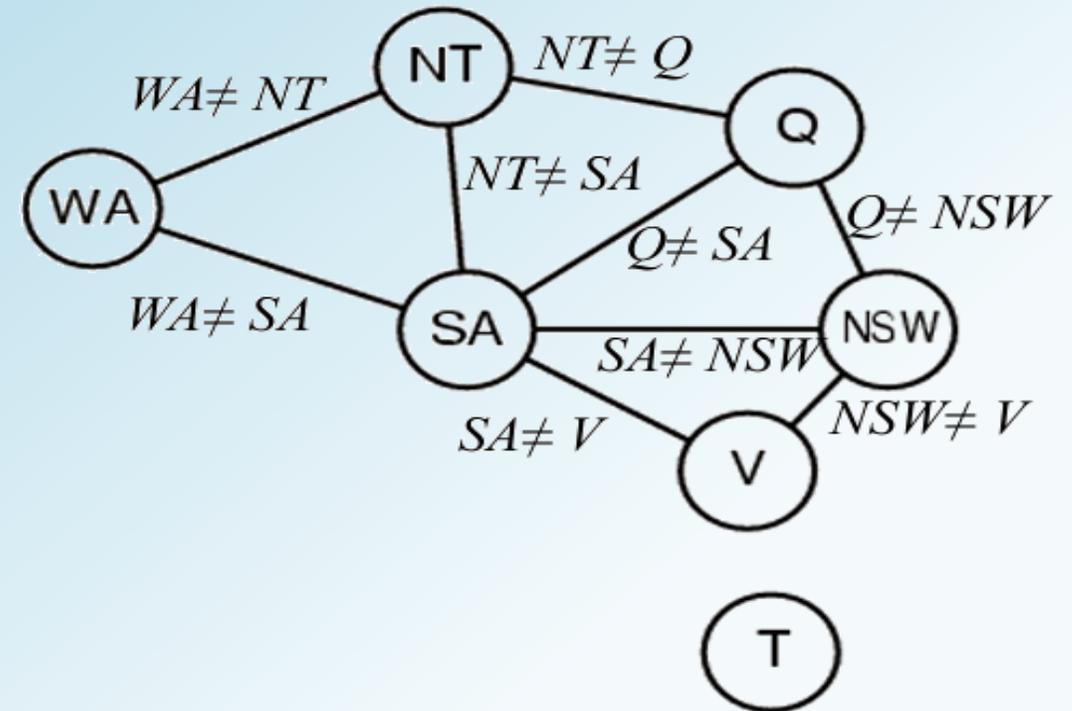
---

- Les variables sont les états :  $V = \{WA, NT, SA, NSW, Q, V, T\}$ .
- Le domaine de chaque variable est l'ensemble de de trois couleurses  $\{V, B, R\}$
- Les contraintes : les régions frontalières doivent avoir des **couleurs différents** : - **WA**  $\langle \rangle$  **NT**, ..., **V**  $\langle \rangle$



# Graphe de contraintes

- Contraintes binaires (entre deux variables) ---On peut visualiser par un graphe de contraintes(SA≠WA)
- ETC



# Approche Simple retour arrière

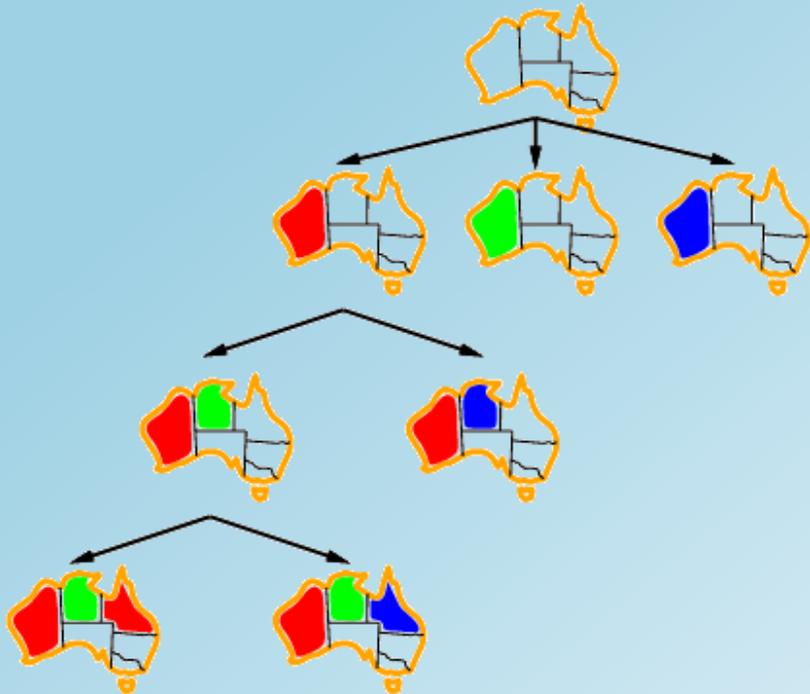
## Backtraking (BT)

---

□ Le retour en arrière est l'action de retourner vers une variable précédant de la variable courante dans la sous-séquence d'instanciation, de modifier la valeur affectée à celle-ci et de poursuivre la résolution par la suite.

1. permettant d'avancer vers une prochaine variable à instancier et permettant d'étendre la sous-séquence et,
2. retour en arrière consistant à choisir la variable vers laquelle sera fait le retour en arrière dans la sous-séquence.

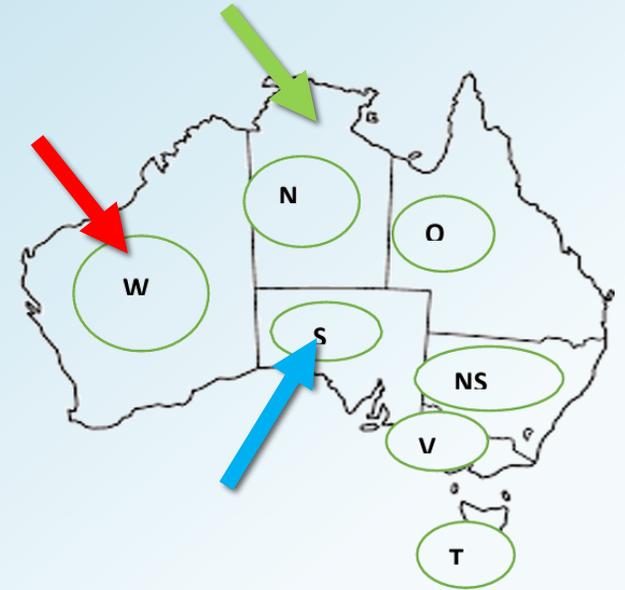
# Backtracking (BT) SEARCHING



```
fonction SRA(A,(X,D,C)) : booléen
début
si A n'est pas consistante alors retourner alors
    retourner FAUX
finsi
si toutes les variables de X sont affectées alors
    retourner VRAI
sinon
    choisir une variable  $X_i$  de X qui n'est pas encore affectée
    pour toute valeur  $V_i$  appartenant à  $D_i$  faire
        si SRA(A  $\cup$   $\{(X_i, V_i)\}$ , (X,D,C)) = VRAI alors
            retourner VRAI
        finsi
    fin pour
    retourner FAUX
finsi
```

# Choisir la prochaine variable

choisir la variable ----**Minimum Remaining Value (MRV) Heuristic** ou **Most Constrained Variable (MCV) Heuristic**.

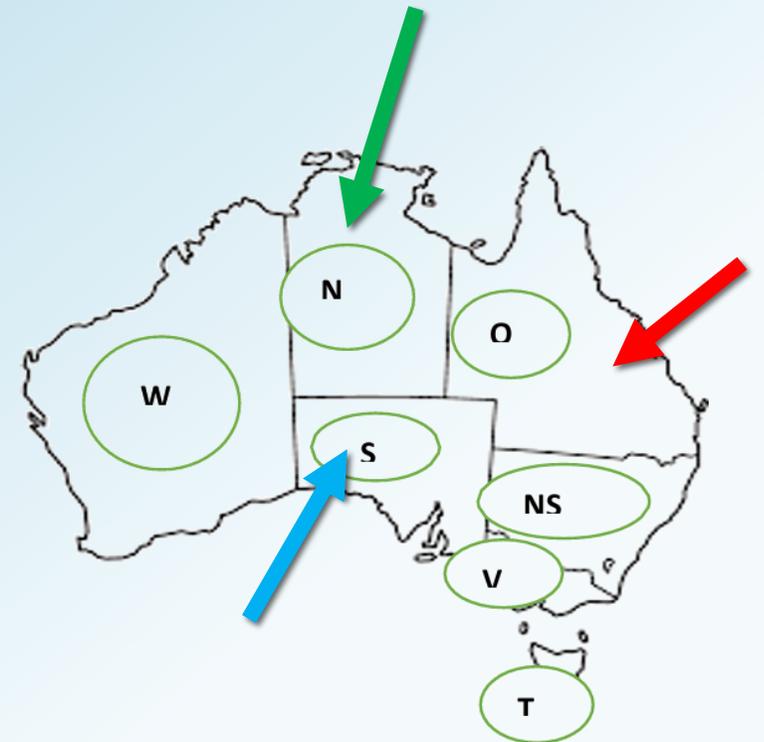


# Choisir la prochaine variable

même nombre de valeurs consistants restantes



- *degree heuristic* (le plus de contraintes).



# solution

---

{WA = Rouge, NT = Vert,  
Q = Rouge, SA = Bleu,  
NSW = Vert, V = Rouge,  
T = Vert}.



---

# Questions