

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

السَّلَامُ عَلَيْكُمْ وَرَحْمَةُ اللَّهِ وَبَرَكَاتُهُ

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Abdelhafid Boussouf University Center of Mila

1^{ère} année Master

Cours Intelligence Artificielle et ses Applications



Chapitre 01

Partie 4

Responsable module

Dr. MEGUEHOUT Hamza



Problèmes Satisfaction des Contraintes

Résoudre certains
problèmes



Les **formulant** comme des **problèmes 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 le **choix des successeurs d'un nœud** est fait par la fonction de transition

Les nœuds du graphe sont « **obscur** » vis-à-vis de la l'algorithme de recherche





Problèmes Satisfaction des Contraintes CSP

Comme **cas particulier** de la **recherche heuristique**

Nœud = **ensemble de variables** avec des **valeurs correspondantes**

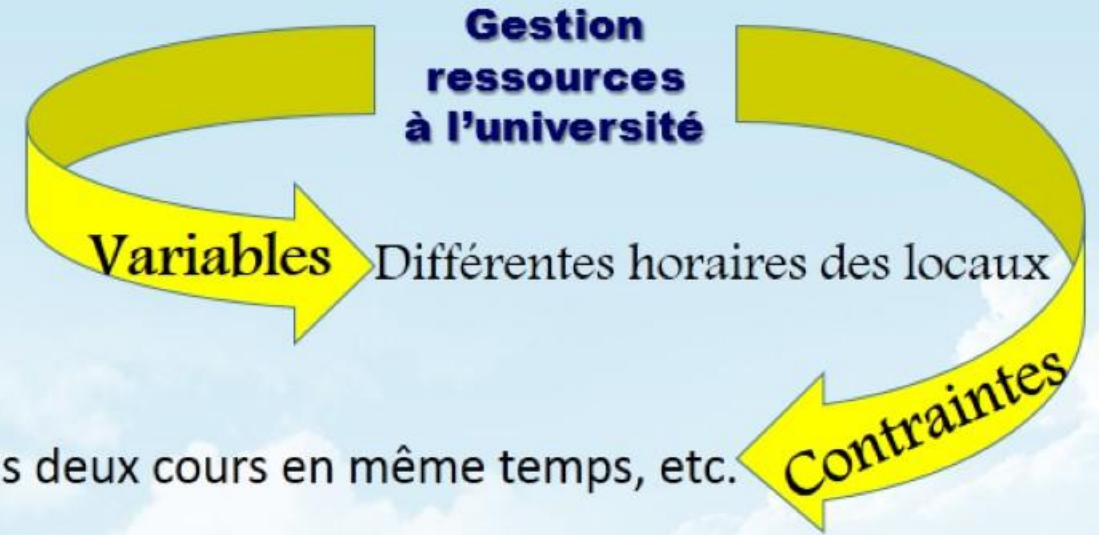
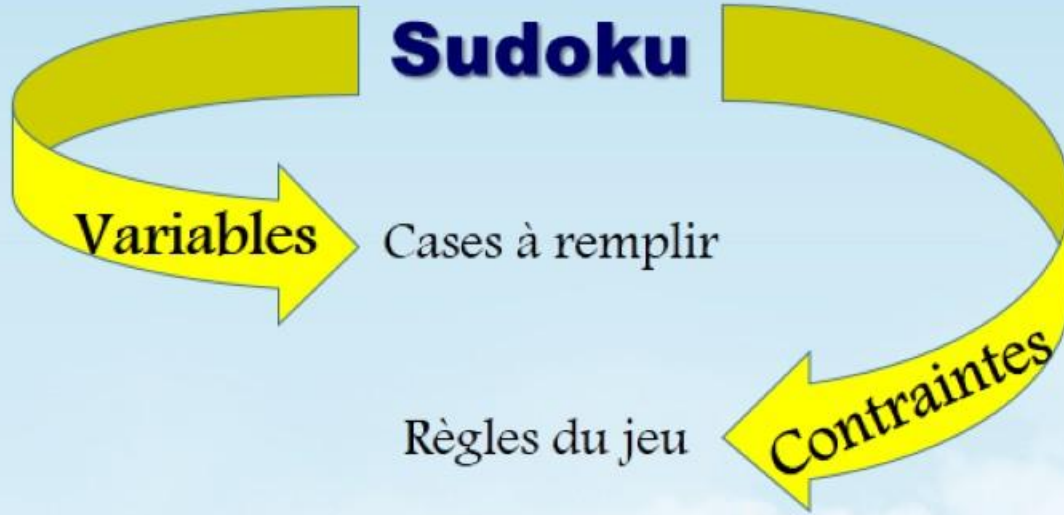
Transitions entre les nœuds **tiennent compte** de **contraintes** sur les **valeurs possibles** des variables

Traduisant un problème **sous forme de satisfaction de contraintes**
On **élimine** la difficulté de **définir l'heuristique $h(n)$** pour notre application





Exemples



Un seul cours au même local à un instant t + l'étudiant n'a pas deux cours en même temps, etc.





Définition formelle

Var = $\{X_1, \dots, X_n\}$ → Ensemble fini de variables (Chaque variable **X** a un **domaine D** de **valeurs possible**)

Con = $\{C_1, \dots, C_M\}$ → Ensemble fini de contraintes (une **contrainte restreint** les valeurs pour un sous-ensemble de variables)

Nœud (état) → Défini par assignation de valeurs à certaines variables ou à toutes les variables

- Assignation **Compatible** → Viole aucune contrainte
- Assignation **Complète** → Concerne toutes les variables
- Solution de CSP → **Compatible** + **Complète**





Définition formelle

$$V = \{X1, X2, X3\}$$

$$D1 = D2 = D3 = \{1, 2, 3\}$$

$$\text{Contrainte} \rightarrow X1 + X2 = X3$$

Solutions



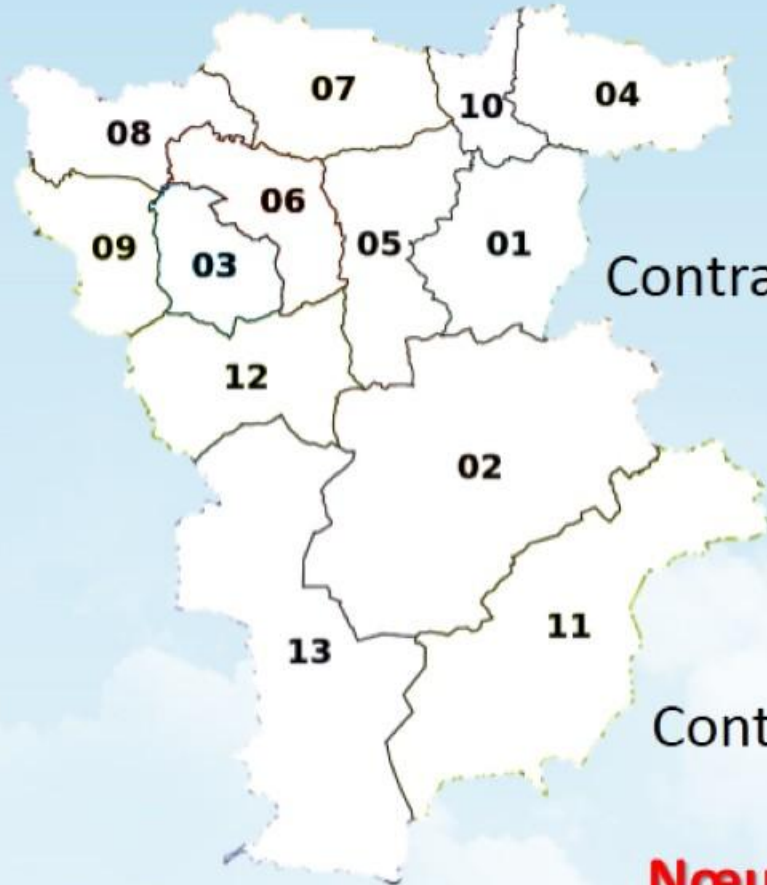


Définition formelle

La carte des daïras de Mila

Utiliser seulement (04) couleurs

Contraintes → deux états frontaliers n'auront pas les mêmes couleurs



$V = \{ \quad \quad \quad \}$

Domaine des variables = $\{ \quad \quad \quad \}$

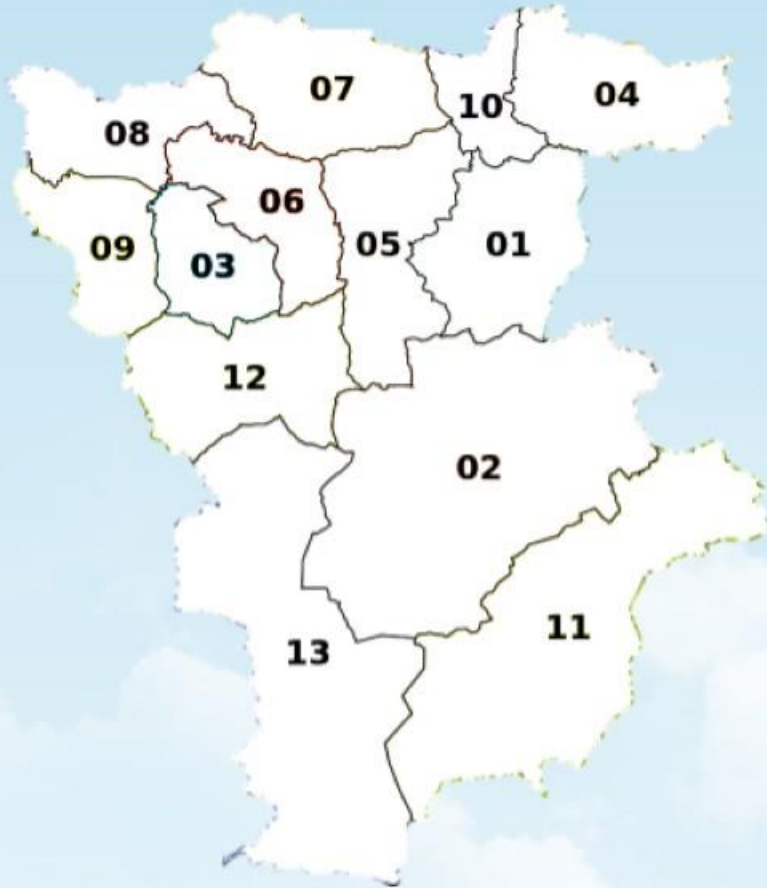
Contraintes =

Contraintes binaires (entre deux variables) → On peut visualiser par un graphe de contraintes

Nœuds = variables (État d'assignation de valeurs ([04 = rouge], [04 = Vert])

Arcs = contraintes entre deux variables





$V = \{01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13\}$

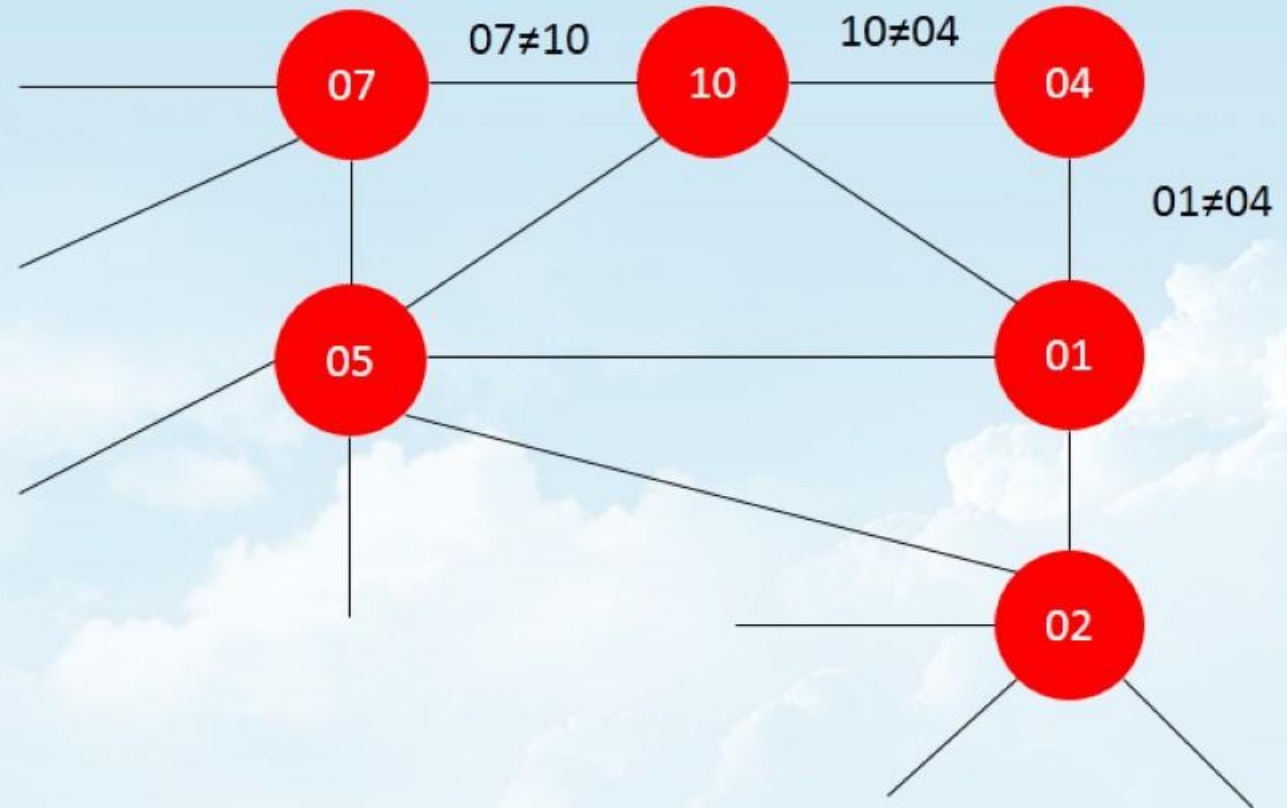
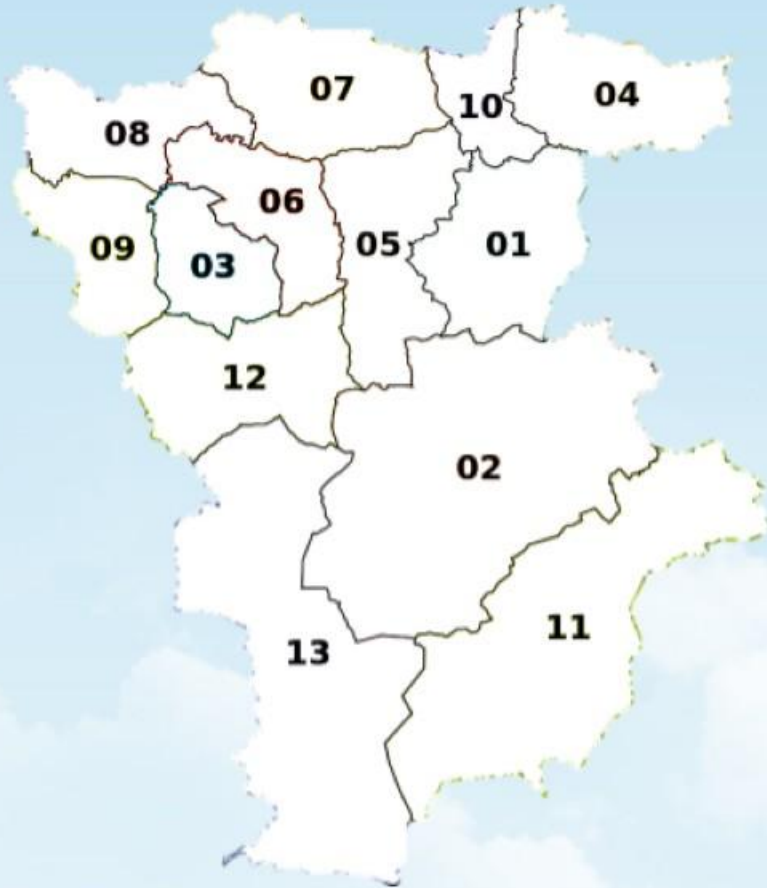
Domaine des variables = { ● ● ● }

Contraintes = **Daïras frontalières** → **Couleurs différentes**



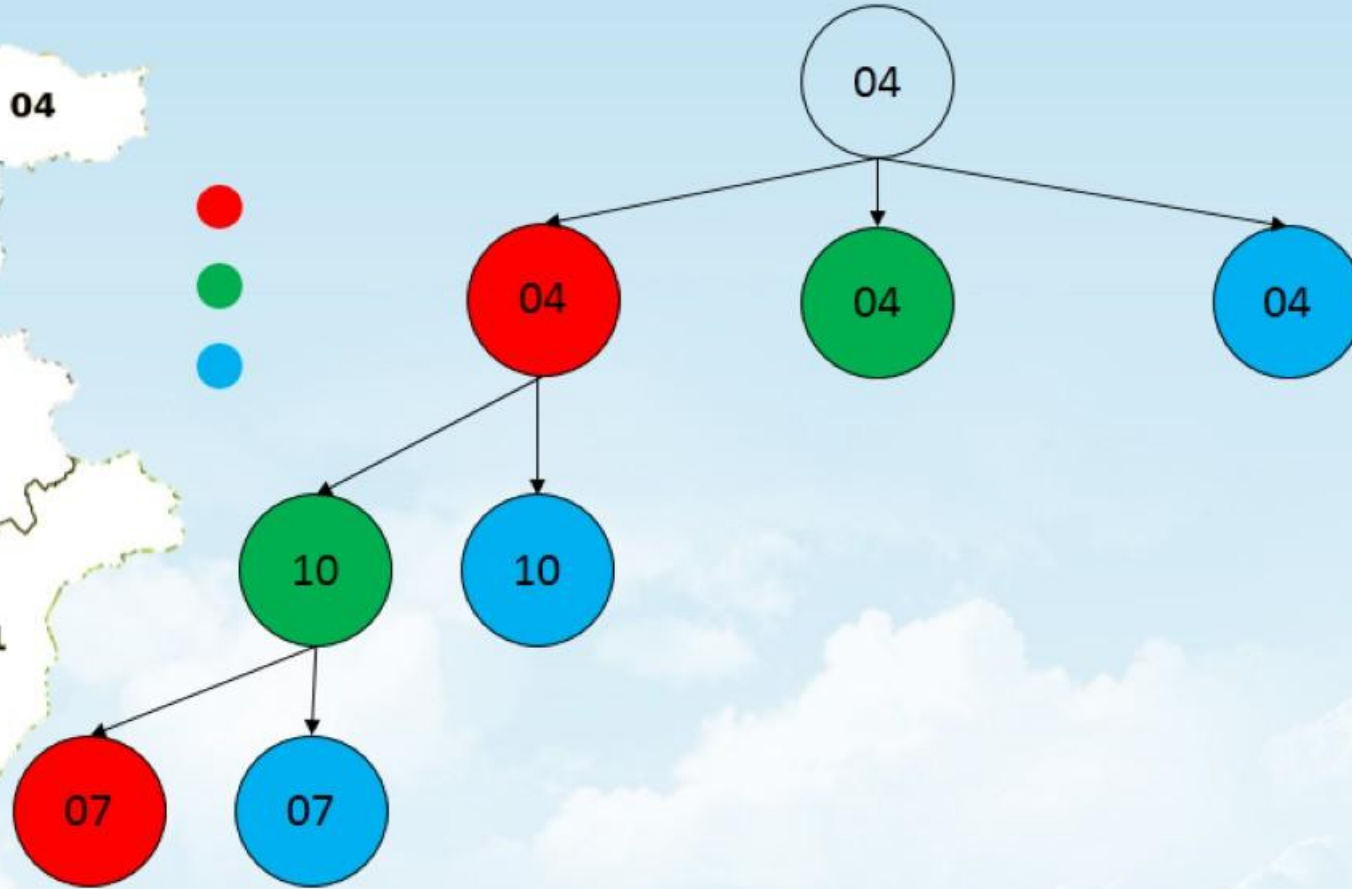
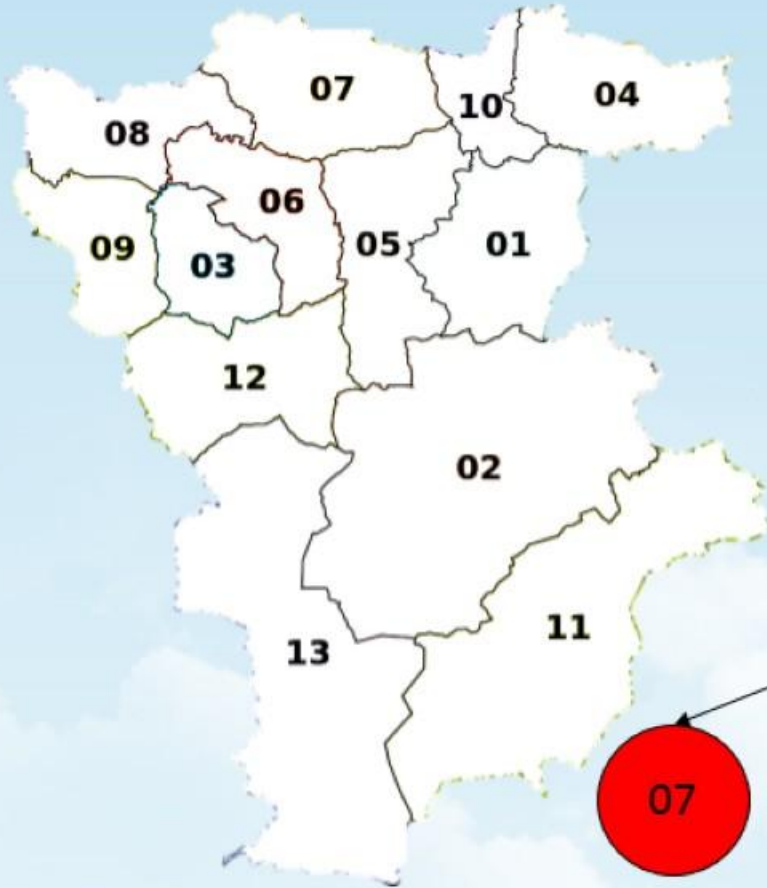


Graphe de contraintes





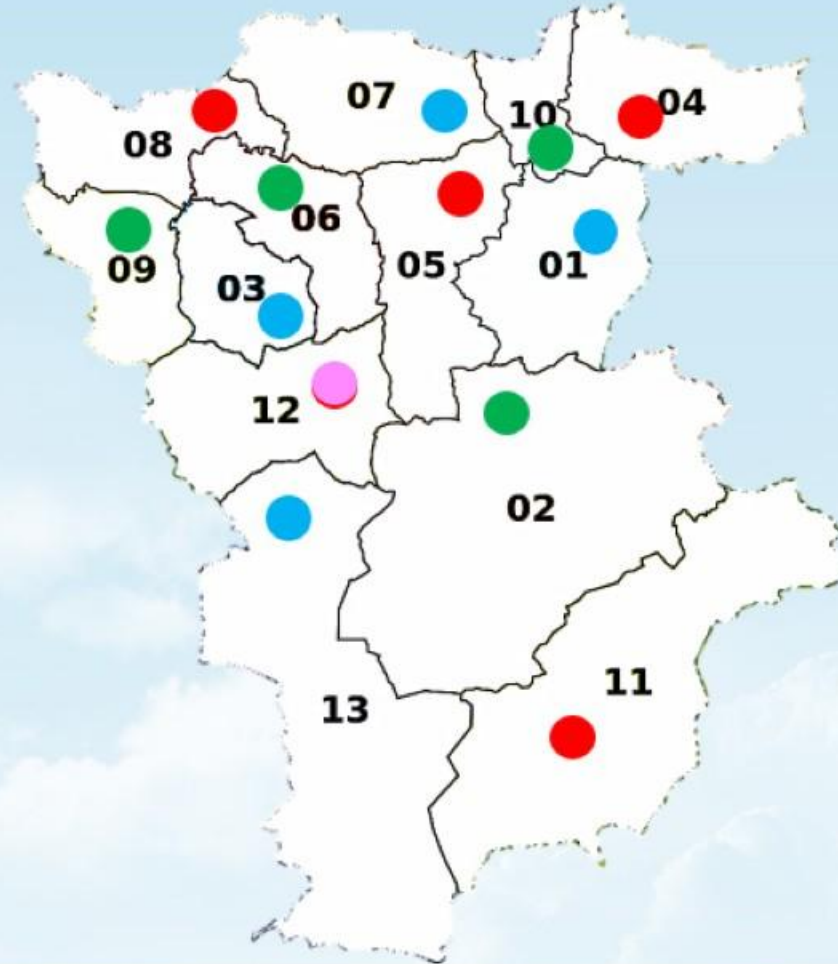
Application Algorithme Backtracking search





Application Algorithme Backtracking search

Variable → moins valeurs compatibles restantes
minimum remaining value



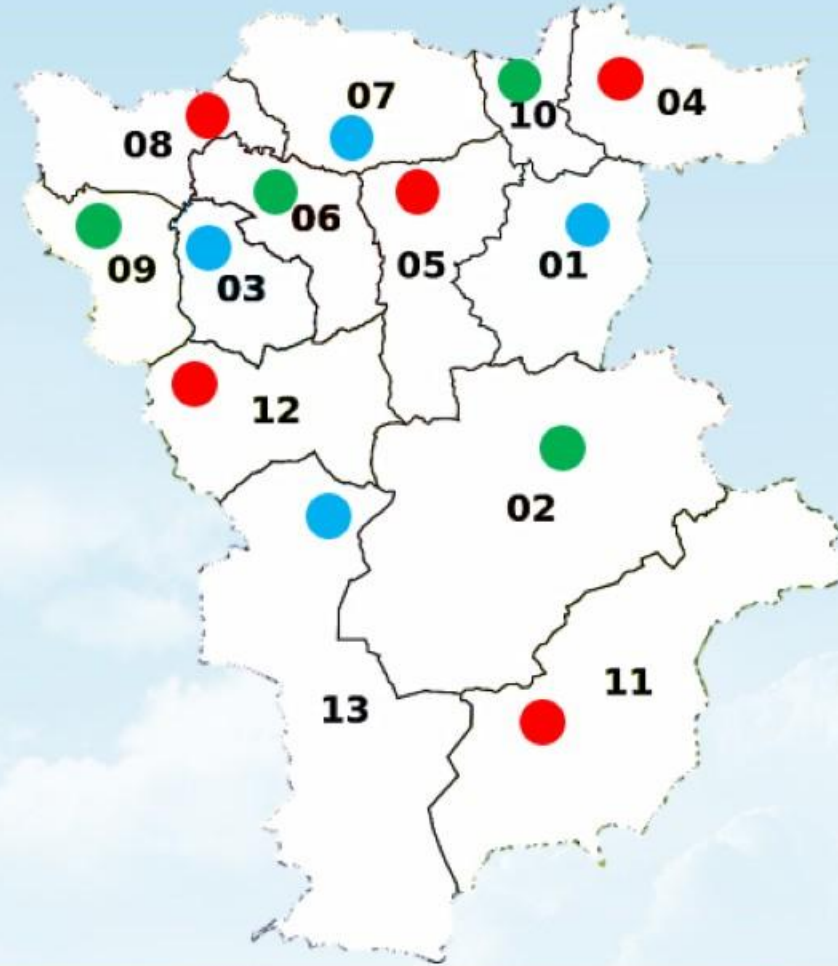


Application Algorithme Backtracking search

Mêmes valeurs compatibles restantes



Choisir plus de contraintes
Degree heuristic



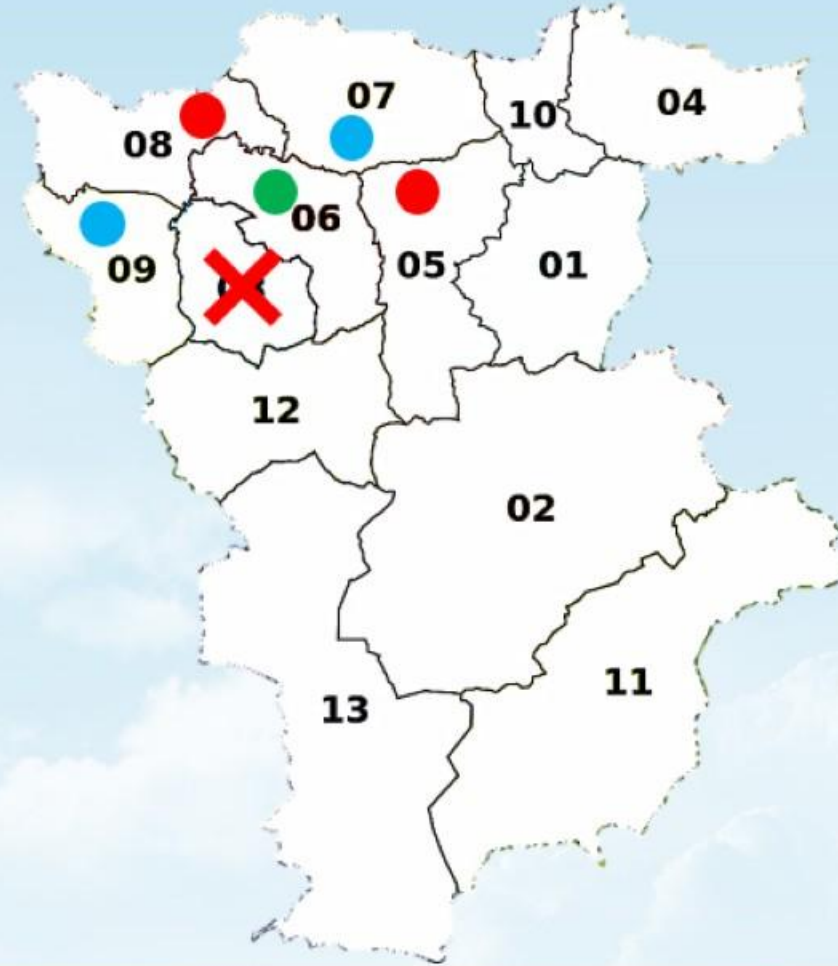


Application Algorithme Backtracking search

Variable X, choisir valeur



Invalide moins de valeurs



*Se réunir est un début, rester ensemble est un progrès,
travailler ensemble est la réussite*

