Résolution des Puzzles “Flow Free” et “n-Queens” avec un Solveur SAT

Bouaita Rayane - Gabriel Trier

February 23, 2025

Abstract

Ce rapport présente la résolution de deux puzzles, *Flow Free* et *n-queens*, en utilisant une réduction en SAT. Nous expliquons la modélisation du problème sous forme de clauses CNF et la résolution à l’aide d’un solveur SAT. L’objectif est de démontrer comment un problème de satisfaction de contraintes peut être converti en une instance SAT et résolu efficacement.

1 Introduction

Le but de ce projet est de résoudre le puzzle *Flow Free / Connecting Dots* par réduction en SAT. Nous abordons également la résolution du puzzle *n-queens* en utilisant un solveur SAT.

1.1 Flow Free / Connecting Dots

L’objectif du puzzle *Flow Free* est de connecter des paires de points de même couleur sur une grille tout en respectant certaines contraintes. Pour cela, nous effectuons une modélisation sous forme de graphe et appliquons une approche SAT. Concrètement, il s’agit de :

- Trouver une affectation de couleurs aux éléments (arêtes) d’un graphe dérivé d’une grille afin de relier, par des chemins continus, des paires de points terminaux de même couleur.
- Remplir entièrement la grille avec des chemins (un pour chaque couleur) qui ne se croisent pas et respectent les règles du jeu.

1.2 n-Queens

Le puzzle *n-queens* consiste à placer n reines sur un échiquier de taille $n \times n$ de manière à respecter les règles suivantes :

- Une seule reine par ligne.
- Une seule reine par colonne.
- Une seule reine par diagonale.

Ce problème est également transformé en une instance SAT et résolu à l’aide d’un solveur SAT.

2 Modélisation du problème Flow Free

2.1 Règles du jeu Flow Free

Dans le jeu **Flow Free**, la grille contient plusieurs paires de points colorés. Chaque paire doit être reliée par un chemin continu, et les règles imposent que :

- **Chaque cellule intermédiaire** (non terminale) doit être traversée par exactement **deux arêtes** de la même couleur, représentant l'entrée et la sortie du chemin.
- **Chaque cellule terminale** (le point coloré) doit être connectée à exactement **une arête** de la même couleur.

2.2 Modélisation en SAT

On représente la grille comme un **graphe** où :

- Chaque cellule est un nœud.
- Chaque connexion possible entre deux cellules adjacentes est une arête.

2.2.1 Définition des variables

Nous définissons :

- e : une arête du graphe reliant deux cellules adjacentes.
- c : une couleur parmi celles disponibles.
- $x(e, c)$: une variable booléenne indiquant si l'arête e est affectée à la couleur c.

2.3 Contraintes de Modélisation en SAT

2.3.1 Contrainte sur les Arêtes

Chaque arête doit avoir exactement une couleur.

Pour chaque arête e, et pour toute paire de couleurs distinctes c_1 et c_2, la clause suivante est ajoutée :

- **Clause :**
$$\neg(x(e, c_1) \wedge x(e, c_2))$$
ou équivalent en CNF :
$$(\neg x(e, c_1) \vee \neg x(e, c_2))$$

2.3.2 Contraintes sur les Nœuds Connecteurs (Non Terminaux)

Chaque nœud non terminal doit être traversé par exactement deux arêtes incidentes, et ces deux arêtes doivent être de la même couleur.

- **(a) Cardinalité :** Pour un nœud v non terminal, considérant l'ensemble $E(v)$ des arêtes incidentes à v, on impose que la somme des variables $x(e, c)$ (pour toutes les couleurs c et pour tous $e \in E(v)$) est exactement égale à 2.

- **(b) Uniformité de Couleur :** Pour chaque paire d'arêtes (e_1, e_2) incidentes à v et pour toute paire de couleurs distinctes c_1 et c_2, on ajoute la clause :
  - **Clause :**
  $$\neg x(e_1, c_1) \vee \neg x(e_2, c_2) \quad \text{si } c_1 \neq c_2$$
  Cela interdit que deux arêtes incidentes utilisées aient des couleurs différentes.

2.3.3 Contraintes sur les Nœuds Terminaux

Chaque nœud terminal doit être connecté par exactement une arête, et cette arête doit être de la couleur imposée par le terminal.

3 Modélisation du problème n-Queens

3.1 Règles du jeu

L'objectif est de placer n reines sur un échiquier $n \times n$ tout en respectant les contraintes suivantes :

- Une seule reine par ligne.
- Une seule reine par colonne.
- Pas plus d'une reine sur une même diagonale.

3.2 Modélisation en SAT

3.2.1 Variables

On associe à chaque case (i, j) de l'échiquier une variable booléenne :

$$x(i, j) = 1 \quad \text{si une reine est placée en } (i, j), \text{ sinon } 0.$$

3.2.2 Contraintes

Soit une grille de taille $n \times n$, où :

- i représente l'indice de la ligne $(1 \leq i \leq n)$,
- j représente l'indice de la colonne $(1 \leq j \leq n)$,
- k est un décalage strictement positif utilisé pour les contraintes de diagonales.

Les contraintes du problème des n-reines sont les suivantes :

1. **Une reine par ligne :** $\forall i, \quad x(i, 1) \vee x(i, 2) \vee \cdots \vee x(i, n)$
2. **Une seule reine par ligne :** $\forall i, \forall j_1 \neq j_2, \quad \neg x(i, j_1) \vee \neg x(i, j_2)$
3. **Une reine par colonne :** $\forall j, \quad x(1, j) \vee x(2, j) \vee \cdots \vee x(n, j)$
4. **Une seule reine par colonne :** $\forall j, \forall i_1 \neq i_2, \quad \neg x(i_1, j) \vee \neg x(i_2, j)$
5. **Une seule reine par diagonale :**
   - **Diagonale descendante :** $\forall i, j, \forall k > 0, \quad \neg x(i, j) \vee \neg x(i + k, j + k)$
   - **Diagonale montante :** $\forall i, j, \forall k > 0, \quad \neg x(i, j) \vee \neg x(i + k, j - k)$

4 Résolution avec un solveur SAT

Les contraintes sont traduites en format CNF (Conjunctive Normal Form) et résolues avec un solveur SAT comme **Gophersat**. Le solveur retourne une affectation des variables qui satisfait toutes les contraintes, et la solution est ensuite interprétée pour afficher le résultat.

5 Comparaison avec d'autres approches

En cours, nous avons étudié des approches basées sur :

- **Programmation par Contraintes (CSP)** : utilise des solveurs dédiés aux contraintes.
- **Programmation Linéaire (PL)** : repose sur des techniques d'optimisation.
- **Programmation Pseudo-Boolean (PB)** : représente les contraintes sous forme d'équations booléennes pondérées.

L'approche SAT est bien adaptée car nous cherchons à **trouver une solution faisable** (plutôt qu'une solution optimale), ce qui correspond exactement au paradigme SAT.

6 Conclusion

Nous avons montré comment les puzzles *Flow Free* et *n-queens* peuvent être transformés en une instance SAT et résolus efficacement à l'aide d'un solveur SAT. Cette approche est une alternative aux techniques classiques de résolution de CSP et se révèle très performante pour ce type de problèmes combinatoires.