



PROJET GRAPHE ET COMPLEXITÉ

Corroller Nathan
Abida Youssef
Rhoulam Khalissa

Rappel du sujet choisi - N reines

Modélisation du problèmes

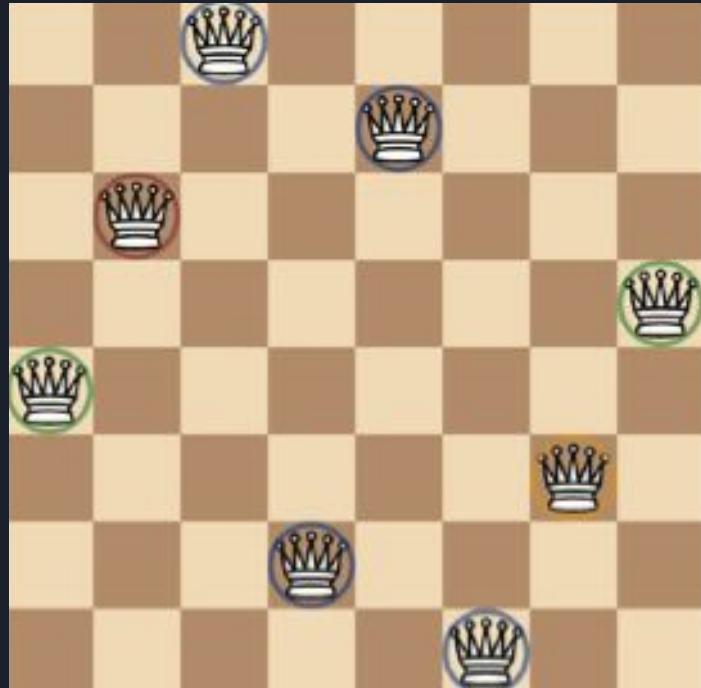
Contraintes :

1. Une seule reine par **ligne**.
2. Une seule reine par **colonne**.
3. Une seule reine par **diagonale**.

Variables : nous utilisons n variables qui représentent les n lignes. La valeur de ces variables représentent la présence de une reine sur la colonne et leur emplacement

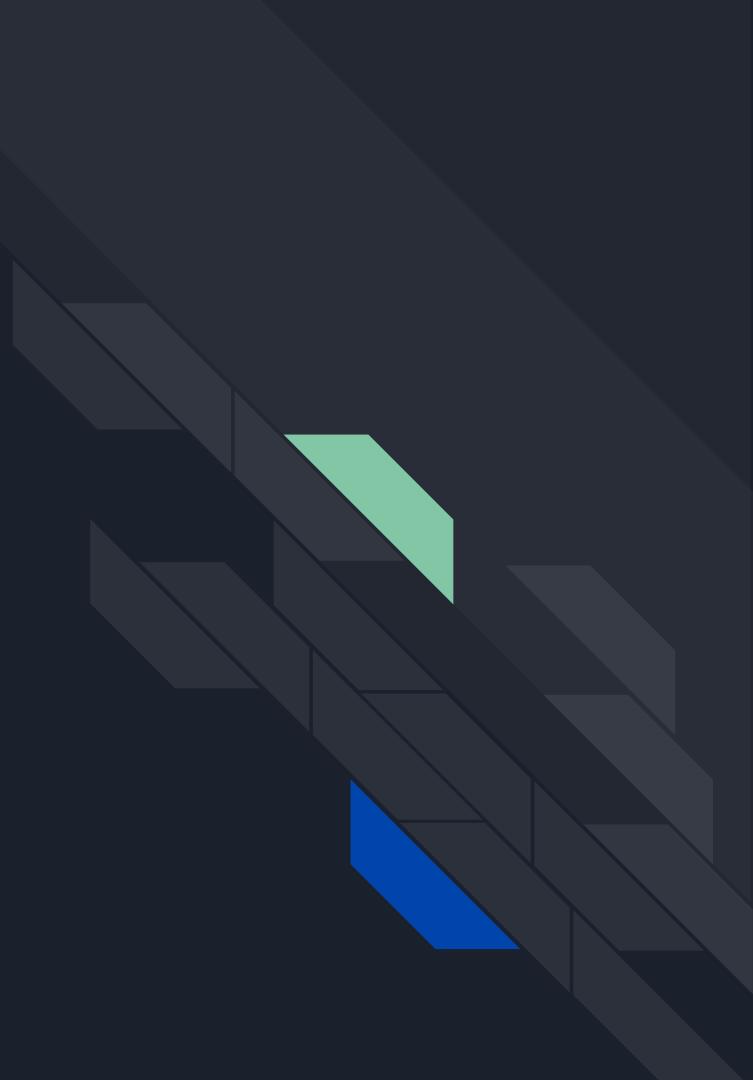
Domaine : chaque variable peut prendre une valeur entre **0** et **n-1** en fonction de où se trouve la reine sur la ligne

Type : C'est un problème de satisfaction de contraintes (CSP) classique



Pour n = 8

Approches complètes



Fixed search fail first

Principe “Fail First”

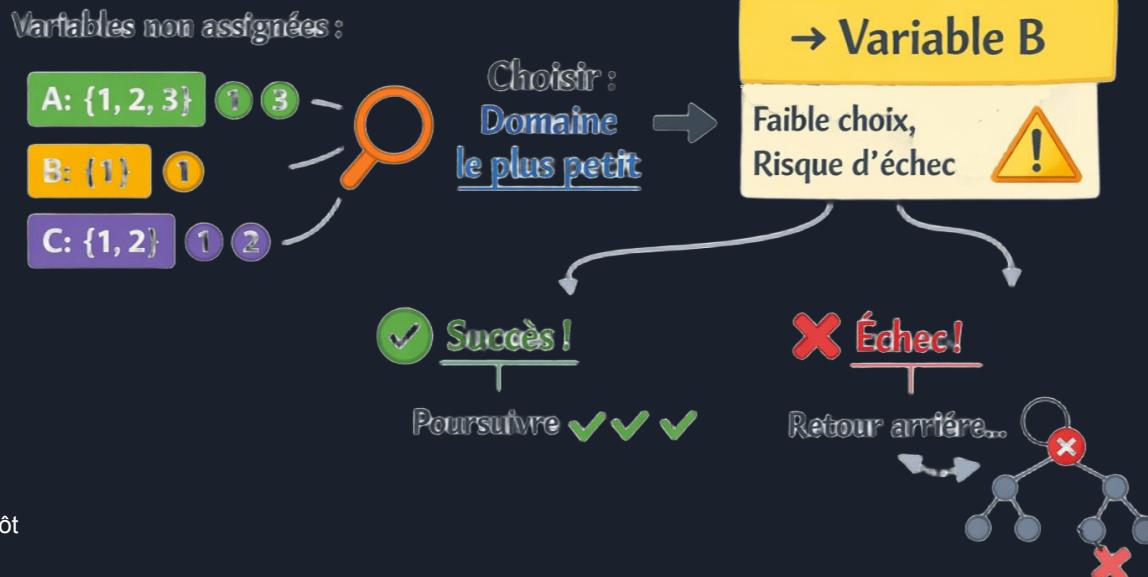
Heuristique MRV (Minimum Remaining Values)

Fail first :
Principe :

On va choisir la variable la plus susceptible d'échouer

- ❖ Variable avec le plus petit domaine
- ❖ Heuristique MRV (Minimum Remaining Values)
idée à retenir :

Si une erreur est inévitable, autant la détecter le plus tôt possible.



Fixed search center out

Fixed Search – Center Out :

Principe :

Les variables centrales sont souvent les plus contraintes ou les plus influentes. On a donc une stratégie qui ne dépend pas des domaines

Fixed Search – Center Out est une stratégie d'ordre de recherche où :

- ❖ on commence par les variables **au centre**
- ❖ puis on s'éloigne progressivement vers les extrémités

on utilise un ordre de variables en partant du centre du problème vers l'extérieur, afin de propager les contraintes le plus tôt possible

Principe “Fixed Search – Center Out”

Stratégie d'ordre des variables

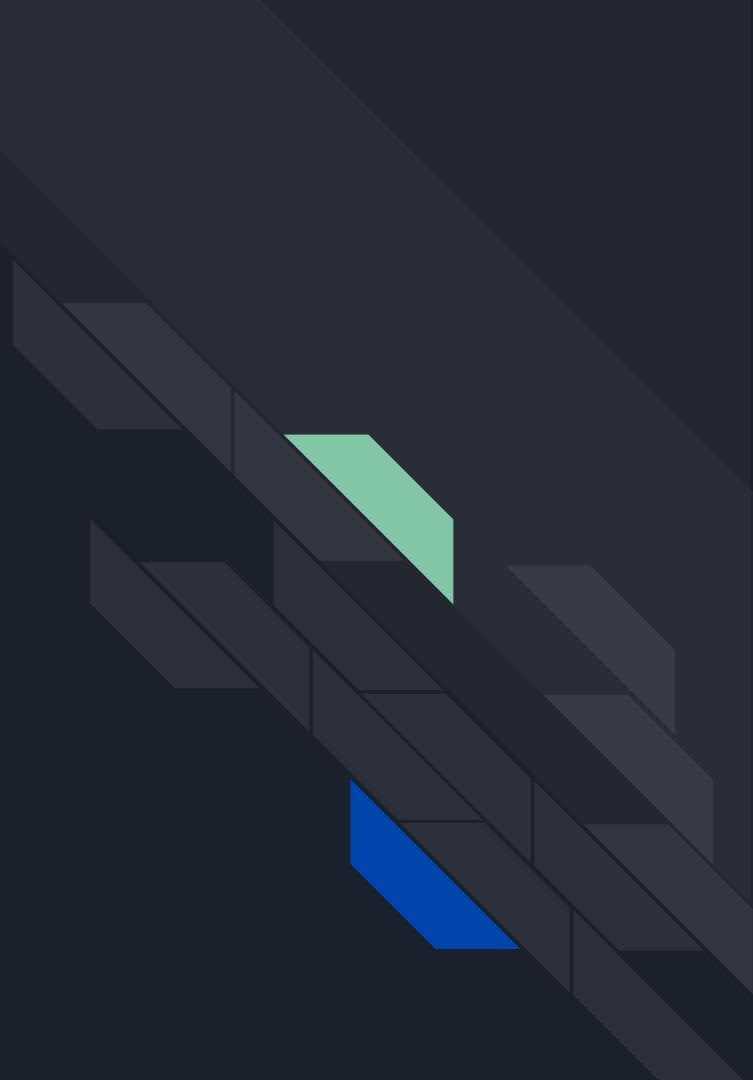


Ordre fixé à l'avance: $X3 \rightarrow X2 \rightarrow X4 \rightarrow X1 \rightarrow X5$

✓ Commencer au centre

✓ S'éloigner progressivement

Approches incomplètes



LNS - Large Neighborhood search

LNS (Large Neighborhood Search) :

Principe :

LNS est une météohéuristique de recherche qui consiste à détruire une grande partie d'une solution, puis à la reconstruire pour explorer un large voisinage.

LNS est utilisé quand :

- ❖ on ne veut/peut pas explorer tout l'espace de solution
- ❖ l'espace des solutions est trop grand
- ❖ une méthode complète serait trop lente
- ❖ on veut rapidement une bonne solution

1. **Solution initiale** : Une solution valide, Trouvée rapidement Pas forcément optimale

2. **Destruction** : On choisit un ensemble de variables et on les libère, 10 %, 30 %, parfois plus, choix aléatoire ou ciblé

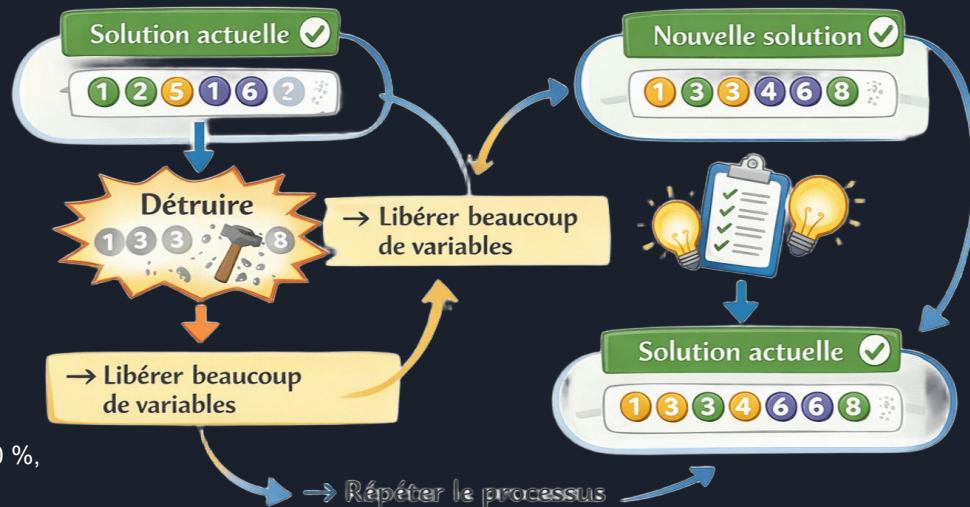
3. **Reconstruction** : Les variables non détruites restent fixées, On réaffecte les variables détruites en respectant toutes les contraintes

4. **Acceptation** : Si la solution est meilleure → acceptée

Parfois acceptée même si elle est un peu moins bonne pour éviter les optima locaux

Principe “Large Neighborhood Search (LNS)”

Méthode météohéuristique d'optimisation



Min-conflict algorithm

Min conflict algorithm:

Principe :

Min-Conflicts est un algorithme de recherche locale qui résout spécifiquement des CSP en minimisant progressivement le nombre de conflits.

Au lieu de :

- ❖ construire une solution pas à pas
- ❖ backtracker en cas d'erreur

Min-Conflicts :

- ❖ part d'une solution complète (souvent invalide)
- ❖ réduit le nombre de conflits localement

Min-Conflicts = corriger les erreurs plutôt que construire sans erreur

1. Générer une affectation complète aléatoire

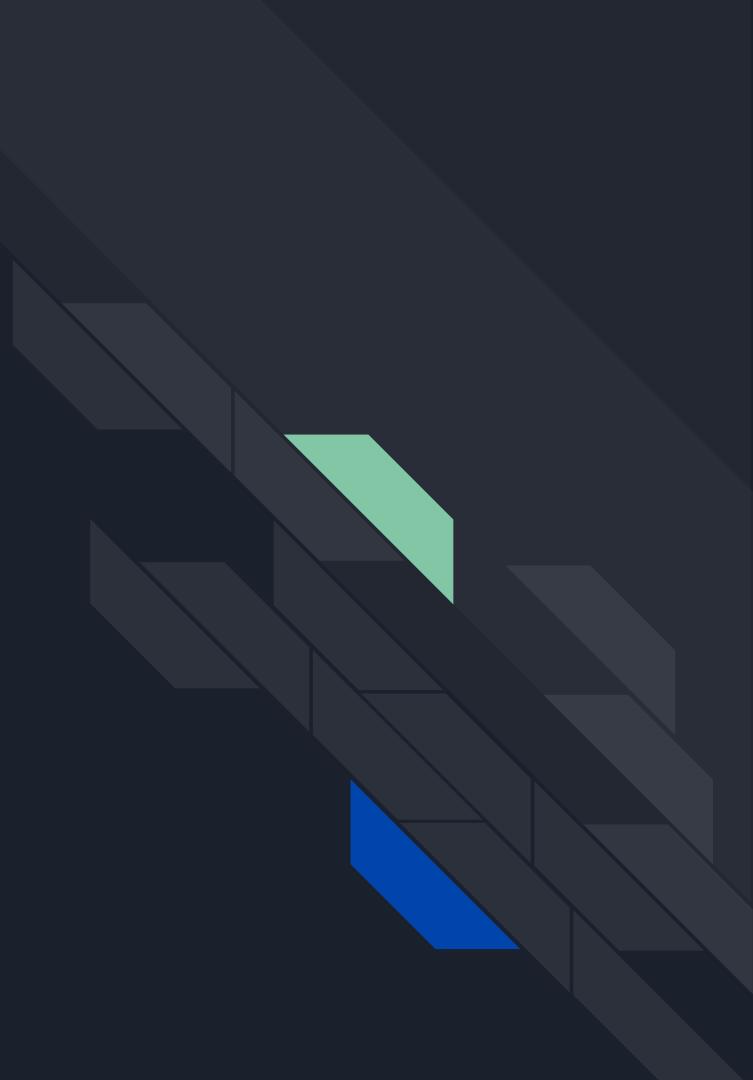
2. Tant qu'il existe des conflits :

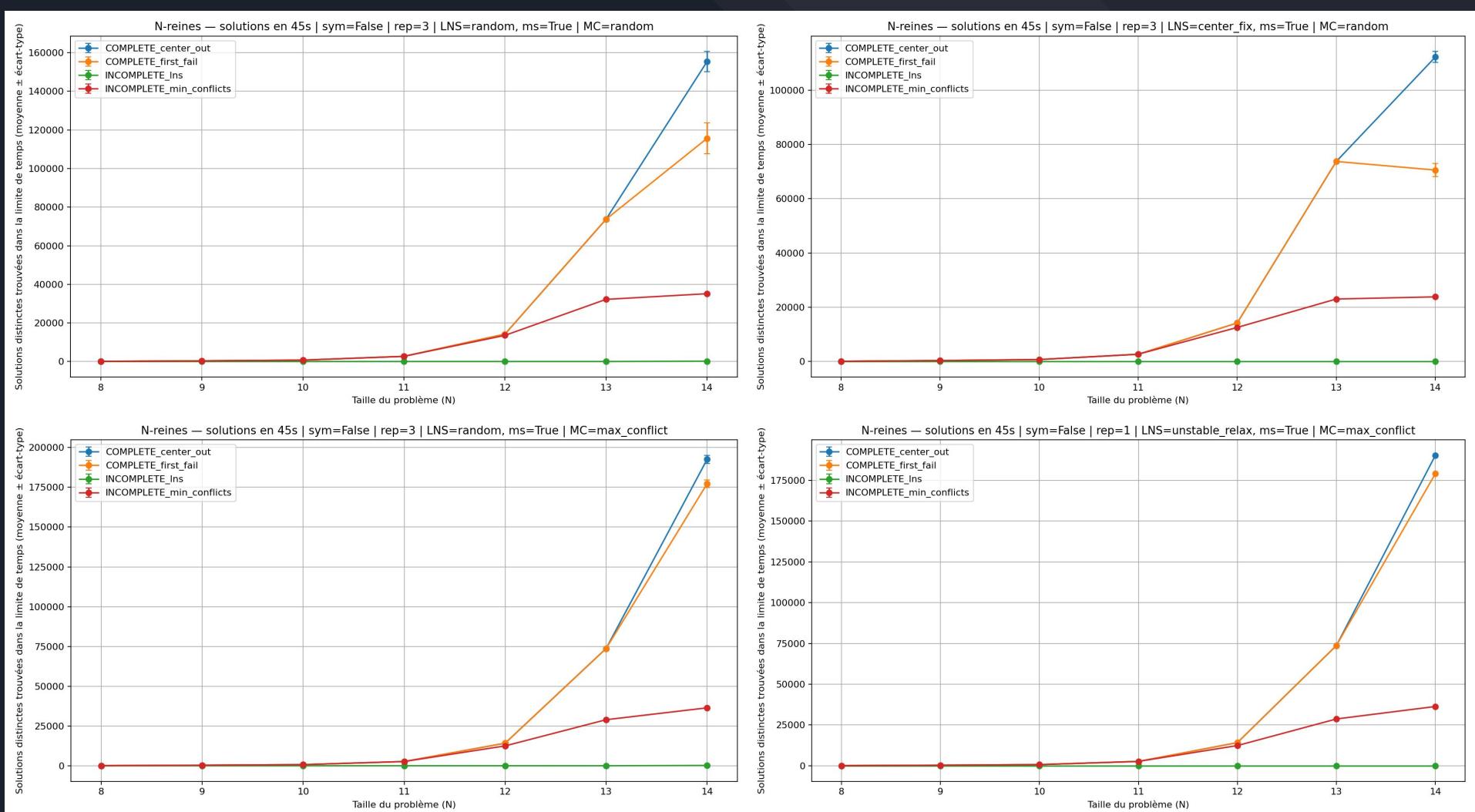
- a) choisir une variable en conflit
 - b) lui donner la valeur qui minimise les conflits
3. Retourner la solution

Algorithme Min-Conflicts



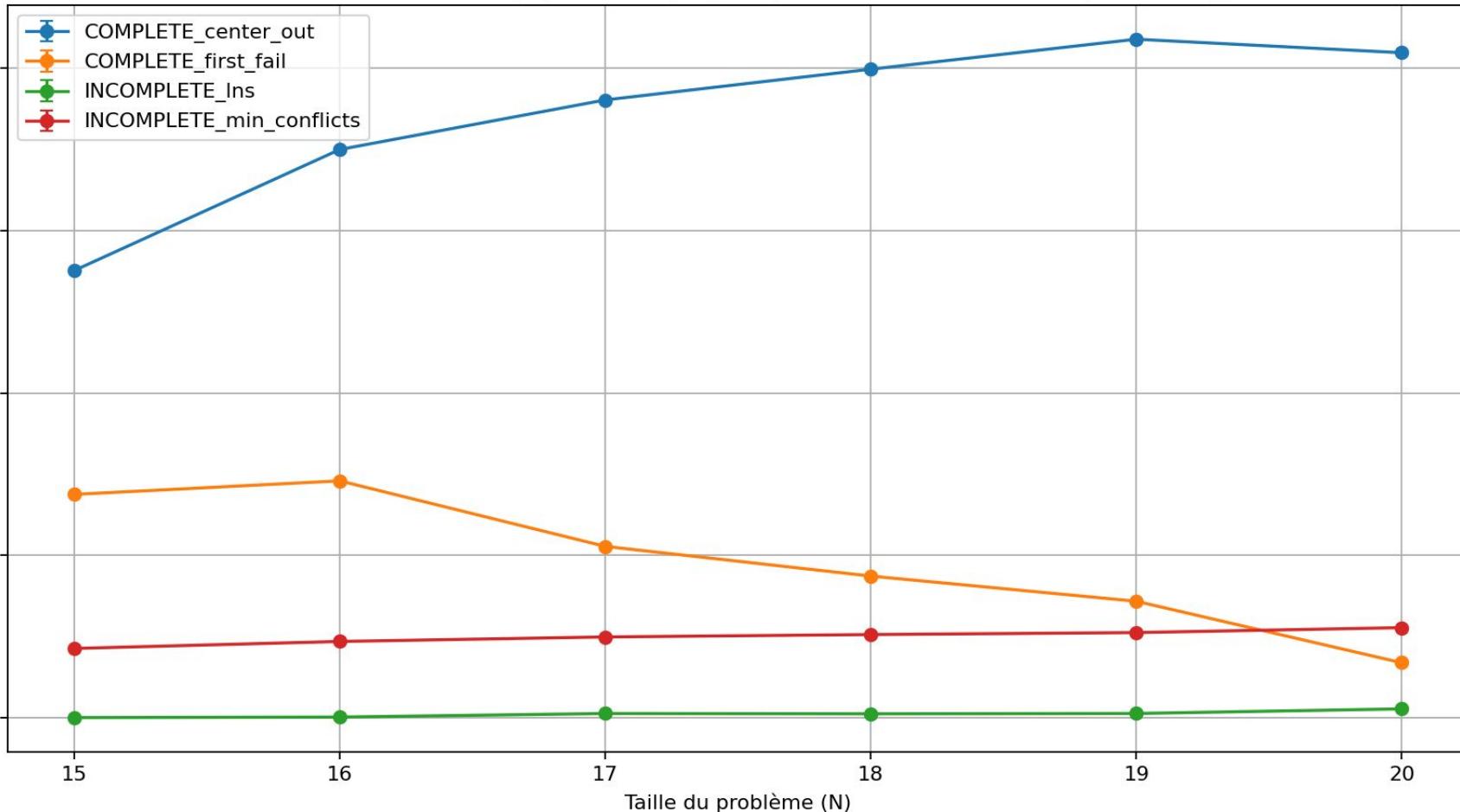
Résultats et comparaisons



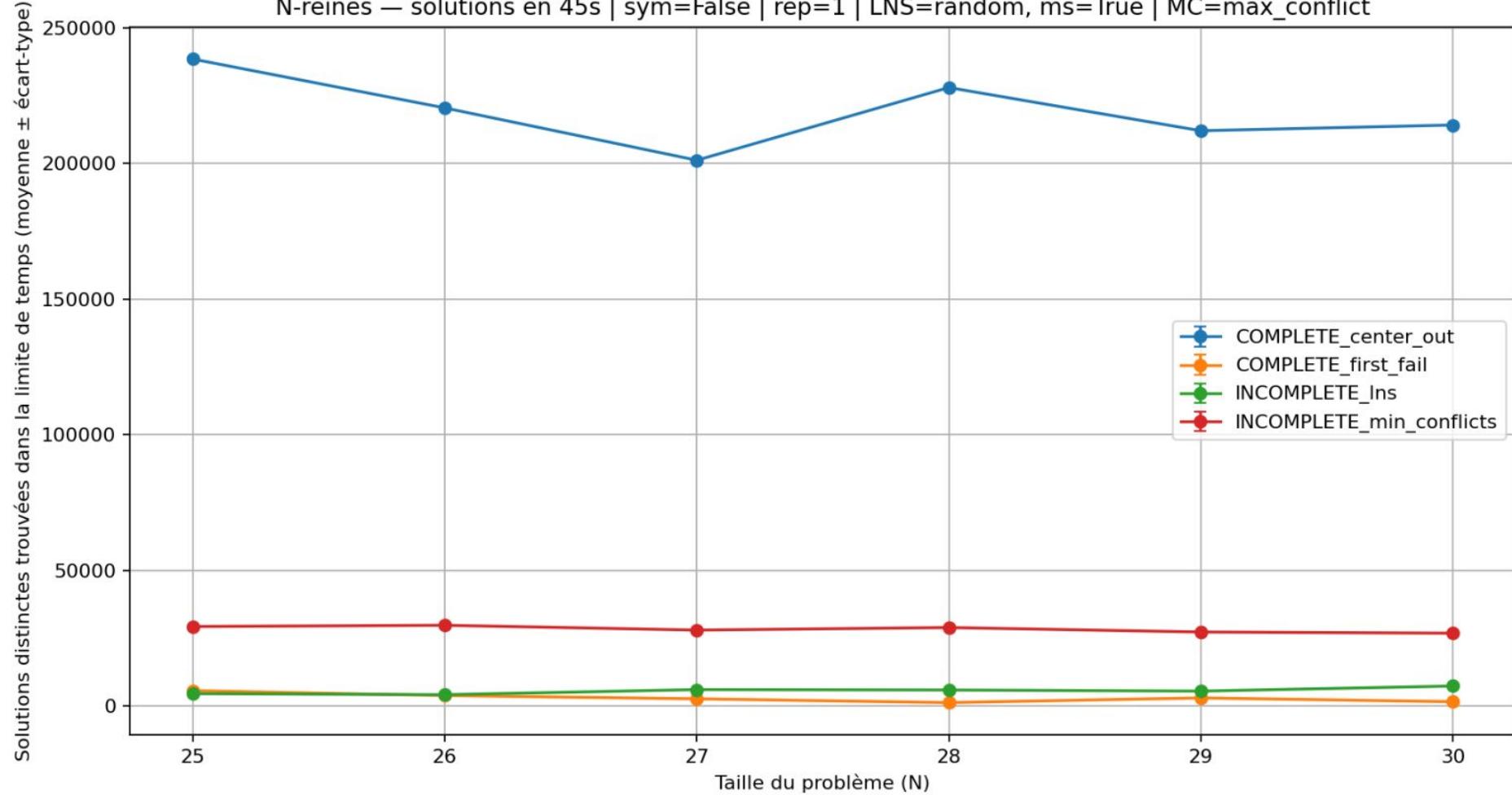


N-reines — solutions en 45s | sym=False | rep=1 | LNS=random, ms=True | MC=max_conflict

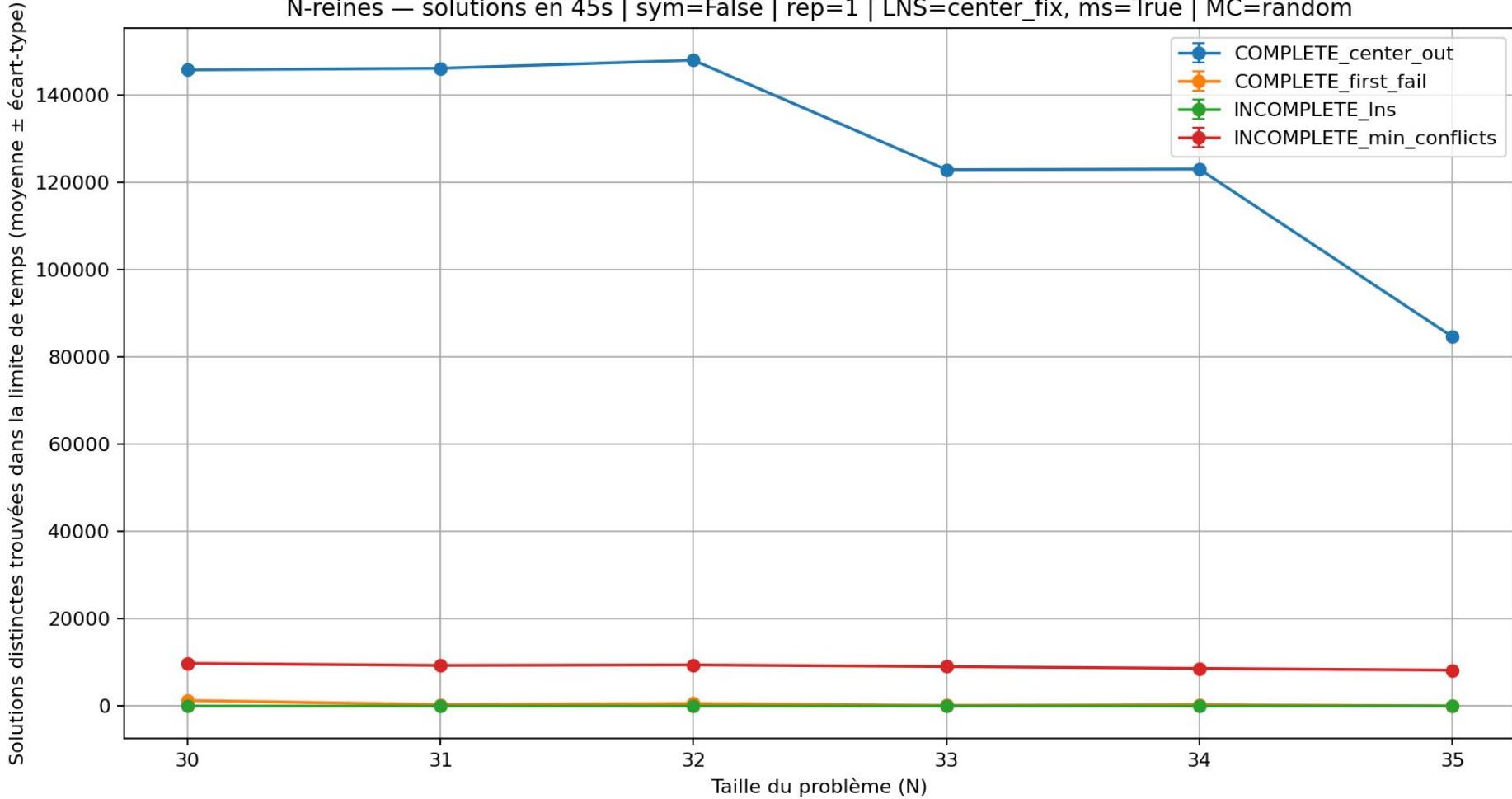
Solutions distinctes trouvées dans la limite de temps (moyenne ± écart-type)



N-reines — solutions en 45s | sym=False | rep=1 | LNS=random, ms=True | MC=max_conflict



N-reines — solutions en 45s | sym=False | rep=1 | LNS=center_fix, ms=True | MC=random



N-reines — Temps 1ère solution (timeout=45s, repeats=1, symmetry_breaking=False, step=2)

