

# CPSolver

## Présentation du projet

Esteve Erwan    Vielzeuf Charles

Projet CPSolver

30 octobre 2025

# Plan

- Résoudre des **CSP** (variables, domaines, contraintes).
- Objectif : cadre modulaire, performant et extensible pour résoudre des instances variées.
- Cas étudiés : **N-Reines, Coloration de graphique, Job Shop Scheduling (JSSP)**.

# Architecture du code

- `core/` : variables, domaines, contraintes
- `parser/` : lecture d'instances
- `algorithms/` : propagation (ex. AC-3), routines
- `solver/` : orchestration de la recherche
- `strategies/` : heuristiques (MRV, degré, LCV, etc.)
- `io/` : sorties, statistiques

# Heuristiques et techniques implémentées

- Propagation : **AC-3**
- Pruning : **Forward checking**
- Choix de variable : **MRV**, heuristique de degré, aléatoire
- Choix de valeur : **LCV**, ordre lexicographique, aléatoire

- Comparaison de 4 configurations : NoAC\_NoForward, NoAC\_Forward, AC\_NoForward, AC\_Forward.
- Mesures : temps, nœuds explorés, backtracks.
- Instances :  $n = 4..10$ .

# N-Reines — exemples de résultats (extraits)

Config	Temps (s)	Nœuds	Backtracks
8 NoAC_NoForward	0.082	15720	2056
8 NoAC_Forward	0.013	1360	1068
8 AC_NoForward	0.656	15720	2056
8 AC_Forward	0.166	1210	1008

- Forward checking réduit drastiquement les nœuds explorés.
- AC-3 aide parfois mais son coût peut dominer pour petits N.
- Statistiques globales : min nœuds = 16, max = 348150, médiane 1052.



- Instances classiques (ex. myciel\*, games120, anna, david,...)
- Test sur différentes valeurs de  $k$  (nombre de couleurs)
- Mesures : statut, solutions, temps (ms), nœuds

Instance	k	Temps (ms)	Nœuds
anna	11	1445	138
david	11	537	87
myciel5	6	21	47
myciel6	7	333	95

- Variable par opération = date de début.
- Contraintes de précédence (opérations d'un job).
- Contraintes de non-chevauchement sur machines.
- Objectif habituel : trouver faisabilité pour un horizon  $K$  ou minimiser le makespan.

# JSSP — benchmark (instance ft06)

K	Statut	Temps (s)	Nœuds
58	Timeout	600	1,021,412
61	Timeout	600	1,020,618
64	First sol.	30.252	103,011
67	First sol.	2.359	1,103
70	First sol.	2.641	1,173

- Trade-off propagation vs coût (AC-3 utile mais pas systématiquement).
- Heuristiques (MRV, LCV) importantes pour réduire l'arbre de recherche.
- Modularité permet d'expérimenter rapidement de nouvelles stratégies.
- Optimisations futures : contraintes globales spécialisées, meilleure gestion des domaines temporels.

# Conclusion

- CPSolver : implémentation modulaire, performante sur petites et moyennes instances.
- Prochaines étapes : contraintes globales, nouvelles heuristiques et optimisation des coûts de propagation.

Merci — questions ?