

PROJET DE SATISFACTION DE CONTRAINTES

M2 MASTER INFORMATIQUE – 2021 / 2022

Travaux sur le jeu Nonogram

Auteur :
David SARMIENTO
Master IAD

Encadrant :
Mr. Bessière CHRISTIAN

16 février 2022



Le Nonogram (en noir et blanc) est un jeu japonais dont l'objectif est de colorier les cases de la grille donnée selon les indications se trouvant sur les lignes et les colonnes. Généralement, le résultat est un dessin plus ou moins "pixélisé" selon la taille de la grille. Les indications sur les lignes et les colonnes sont intuitives :

- "8" - qu'il y a 8 cases à colorier consécutivement.
- "3 2 1" - signifie qu'il y a 3 groupes de cases à colorier de taille respective et dans l'ordre 3, 2, 1 et avec au moins 1 case d'espace entre chaque groupe.

Pour aider le joueur, en plus de la possibilité de colorier une case, il peut aussi mettre une croix pour signifier qu'il est impossible de colorier la case.

La première étape de ce projet est d'implémenter le jeu et de pouvoir en faire un affichage acceptable. La deuxième est de faire un vérificateur si les contraintes des indications sont respectées. Pour cela, il y a une fonction qui vérifie la cohérence de la colonne ou ligne donnée et enfin un agrégateur qui utilise cette fonction sur l'ensemble des indications. Enfin la dernière étape est l'implémentation d'un solveur. Pour cela il y a tout d'abord le solveur naïf qui utilise l'approche *back-track*. Simple à implémenter mais très lent en performance. En effet, sur une grille de taille $n \times m$ la complexité est de $2^{n \times m}$. Par la suite, un deuxième solveur plutôt heuristique qui se rapproche de *AC*. Il utilise le vérificateur pour mener à bien sa tâche.

- A. Au début du jeu, le solveur colorie d'abord les cases qui le sont obligatoirement. Comme la grille est de taille finie, une indication dont la somme, en prenant en compte les espaces, qui dépasserait la moitié de la taille possible d'une LC (ligne ou colonne) implique au moins un coloriage sur cette dernière. Ceci est fait par l'union de la possibilité en commençant l'indication par le début d'une LC et sa fin.
- B. Le solveur propage par la suite les nouvelles informations obtenues : la continuation d'un coloriage parce que le groupe colorie au moins des cases supplémentaires, le marquage impossible des cases du fait des espacements considérés comme sûrs ou juste la complétion d'une LC.
- C. La grille est ainsi mise à jour, on réitère A et B jusqu'à ce que le Nonogram soit résolu.
- D. Il se peut que cette partie ne soit pas implémentée, on ne traitera alors que des Nonograms où il n'y a pas à "deviner". Dans le cas où A et B ne modifient pas la grille, le solveur doit ainsi choisir une LC qui a le moins de choix possible pour effectuer rapidement le solveur naïf. Une fois débloqué, il reprend normalement et fait appel au solveur naïf dans des cas similaires.

L'ordre est bien sûr important pour le solveur naïf. Il aurait été intéressant d'utiliser la propagation B pour parvenir plus rapidement à la solution bien que cela n'ait pas grand impact au début. Le solveur heuristique est très rapide. Une difficulté sur le projet est de pouvoir se rapprocher aisément des modèles vus en cours.

Pour obtenir les résultats il suffira de lancer en ligne de commande "python Main.py" et les résultats s'afficheront.