Université de Caen Normandie
Département Mathématiques & Informatique
L2 informatique, 2020
Unité UE2

Sécurité et aide à la décision

Jeu d'infection

Grégory Bonnet

---

1 Jeu d'infection

Considérons le jeu suivant :

Soit une grille de $N \times M$ cases. Chaque joueur – Blanc et Noir – débute la partie avec un pion, respectivement en bas à droite et en haut à gauche. L'objectif du jeu est de posséder en fin de partie le plus de pion possible, la partie se terminant lorsqu'un des joueurs ne dispose plus de pion, ou si aucun joueur ne peut faire de coups légaux. À chaque tour, le joueur actif choisit un pion et effectue une des actions suivantes avant de passer la main à son adversaire :

- Dupliquer le pion en en plaçant un nouveau sur une case libre à une distance de un dans une des quatre directions cardinales (haut, bas, gauche ou droite). Tout pion adverse en contact avec ce nouveau pion est transformé en pion du joueur.
- Déplacer le pion sur une case libre à une distance de deux dans une des quatre directions cardinale ; ce mouvement permet de sauter par-dessous un autre pion (quel qu'en soit le propriétaire).

Implémentez dans un langage de programmation vu en cours (Python ou Java) le jeu présenté ci-dessus. Pensez à implémenter une représentation de l'état du jeu (grille, pions, joueur actif), une fonction qui calcule combien de pion dispose un joueur, une fonction qui retourne tous les coups possibles d'un joueur donné, une fonction qui prend un coup en entrée un coup et génère un nouvel état.

2 Algorithmes de recherche

Écrivez une implémentation de (1) `minmax` ou `negamax` et (2) un élagage `alphabeta` (ou sa version `negamax`). Spécifiez à haut niveau (interfaces) les objets représentant le jeu et vous prendrez garde à bien obtenir le coup qui doit être joué.

Considérez la fonction d'évaluation suivante : la proportion de pions que possède le joueur par rapport à son adversaire. L'évaluation du joueur blanc est donc $\frac{N_B}{N_B+N_N}$ où N_B est le nombre de pions du joueur blanc et N_N le nombre de pions du joueur noir. Lévaluation du joueur noir est donc $\frac{N_N}{N_B+N_N}$.

Implémentez un compteur pour obtenir le nombre de nœuds explorés au cours de la partie.

3 Expérimentations

Votre programme doit pouvoir être exécuté en lui passant six paramètres :
- la longueur et la largeur de la grille,
- le nombre de coup d'avance donné au joueur blanc,
- la profondeur de raisonnement du joueur blanc,
- la profondeur de raisonnement du joueur noir,
- l'utilisation ou non d'un élagage `alphabeta`.

Réaliser les deux expérimentations suivantes.

(1) En considérant que la profondeur de raisonnement est la même pour les deux adversaires et en fixant la taille de la grille, tracez des courbes représentant le nombre de nœuds explorés par `minmax` puis `alphabeta` selon la profondeur de raisonnement.

(2) Identifiez l'intérêt du raisonnement en profondeur par rapport à des coups d'avances. Fixez une taille de grille et une profondeur de raisonnement pour les deux joueurs. Expérimentez en donnant des coups d'avance au joueur blanc et en augmentant la profondeur pour le joueur noir. Indiquez dans un tableau à deux dimensions (coups d'avance du joueur blanc, surplus de profondeur du joueur noir) si le joueur noir gagne ou perd.

4 Rendu du travail

Le TP devra être rendu pour le lundi 2 mars 9h00 au plus tard sur eCampus : `https://ecampus.unicaen.fr/mod/assign/view.php?id=245034`. Il s'agira d'une archive (ZIP, TAR, GZ au choix) contenant :
- votre code source (compilable ou exécutable sans les scories laissées par un IDE).
- un (éventuel) fichier `readme.txt` afin d'indiquer la marche à suivre.
- un court rapport au format PDF présentant les résultats d'expérimentation.

Les noms des deux membres du binôme doivent être indiqués dans le rapport et le `readme.txt`. La notation reposera sur (1) l'implémentation des règles du jeu, `minmax` et `alphabeta` ; (2) les expérimentations réalisées ; (3) la propreté et la lisibilité du code.