

Intelligence Artificielle

TP : stratégies de jeux

1 Le jeu des bâtonnets

Nous allons implémenter en Python3 le jeu des bâtonnets et l'algorithme Minimax. Le principe du jeu est simple :

- 27 bâtonnets sont disposés sur la table ;
- chacun son tour, chaque joueur prend 1, 2 ou 3 bâtonnets ;
- celui qui prend le dernier a perdu.

Pour ce jeu assez simple, l'algorithme minimax est utilisable sans limitation de profondeur. Nous représenterons l'état de la partie avec une variable entière `N` contenant le nombre de bâtonnets restants sur la table.

1. Téléchargez le programme `batonnets.py` sur ARCHE, lancez-le pour vérifier qu'il permet bien de jouer et regardez son code pour comprendre le fonctionnement global.
2. Complétez la fonction `ordiJoue(N)` pour calculer le meilleur coup possible en utilisant les fonctions `valeurMax` et `valeurMin` au lieu de choisir toujours 1 bâtonnet.
3. Complétez les fonctions `valeurMax` et `valeurMin` pour calculer la valeur minimax d'un nœud et la fonction `ordiJoue(N)` pour afficher en plus du coup joué `je vais gagner` ou `je vais perdre` selon la valeur calculée pour le coup joué.
4. Testez le programme. Combien de temps met-il pour jouer le premier coup ? Pouvez-vous gagner s'il joue le premier coup ?
5. Accélérez sa réflexion en implémentant la technique alpha-beta dans deux nouvelles fonctions `valeurMaxAlphaBeta` et `valeurMinAlphaBeta` que vous utiliserez au lieu de `valeurMax` et `valeurMin`. Vous pouvez les créer en copiant les fonctions de départ, puis en les modifiant. Vous devrez notamment ajouter deux paramètres d'entrée (`alpha` et `beta`). Quel est maintenant le temps de calcul pour le premier coup ? Comparez le nombre de nœud explorés entre les deux versions.
6. Testez (si possible) l'exécution des deux versions avec `pypy3 batonnets.py` au lieu de `python3 batonnets.py`. Commentaires ?

2 Puissance 4

Nous allons maintenant implémenter l'algorithme Minimax et ses différentes variantes pour jouer au jeu Puissance 4. L'état du jeu sera encodé dans une matrice de 6 lignes et 7 colonnes dans laquelle +1 représente un jeton de l'ordinateur, -1 un jeton de l'adversaire humain et 0 l'absence de jeton.

1. Récupérez le fichier `puissance4.py` sur ARCHE.
2. Lancez le programme `puissance4.py` pour vérifier que tout fonctionne et regardez son code pour comprendre le fonctionnement global.

3. Complétez la fonction `ordiJoue` pour calculer le meilleur coup possible en utilisant les fonctions `valeurMax` et `valeurMin` au lieu de choisir une action au hasard, en ignorant le paramètre `profondeur_max` dans un premier temps.
4. Complétez les fonctions `valeurMax` et `valeurMin` en ignorant le paramètre `profondeur` dans un premier temps.
5. Testez le programme. Combien de temps met-il pour jouer le premier coup ?
6. Vous pouvez arrêter le programme avec Ctrl+C. Donnez une estimation du temps de calcul nécessaire pour jouer le premier coup avec l'algorithme minimax.
7. Nous allons donc implémenter la limitation en profondeur pour éviter ces problèmes. Pour cela, le paramètre `profondeur` des fonctions `valeurMax` et `valeurMin` doit toujours correspondre à la profondeur du nœud évalué. Si cette profondeur correspond à la `profondeur_max`, alors il faut simplement retourner la valeur donnée par la fonction d'évaluation.
8. Testez à nouveau le programme avec une `profondeur_max` de 4. Combien de temps met-il pour jouer le premier coup ? Jusqu'à quelle profondeur maximum peut-on aller tout en conservant des temps de calcul raisonnables ?
9. Complétez la fonction `ordiJoue` pour qu'elle se termine par un affichage adapté à la valeur estimée du coup joué parmi `je vais gagner`, `je vais sûrement gagner`, `je vais perdre` ou `je peux faire match nul`.
10. Pour pouvoir aller plus en profondeur dans la réflexion de l'algorithme, copiez les fonctions `valeurMax` et `valeurMin` dans de nouvelles fonctions et modifiez-les pour implémenter la technique alpha-beta. Vous devrez notamment passer deux paramètres supplémentaires aux fonctions : `alpha` et `beta`.
11. Pensez ensuite à modifier la fonction utilisée dans `ordiJoue` et testez à nouveau le programme. Jusqu'à quelle profondeur peut-on aller raisonnablement avec cette version ?
12. À partir de quelle profondeur le programme vous bat à tous les coups ?