

# Idées d'intelligences artificielles

Pour un jeu à  $n$  joueurs ( $n \in \llbracket 1, 4 \rrbracket$ ), on supposera qu'on est le joueur 1. On notera  $sc_i(s)$  le score du joueur  $i$  dans l'état  $s$ . On note  $sc = sc_1$  notre score. On introduit la mobilité d'un joueur comme le nombre de cases accessibles à partir de l'état  $ss$  et on la note  $m_i(s)$  pour le joueur  $i$ . Ici encore on note  $m = m_1$ .

## 1 | Coup par coup

À chaque tour, on cherche à jouer le meilleur coup parmi tous les coups possibles. On peut définir le meilleur coup de différentes façons : le meilleur coup peut être celui qui

- maximise mon score
- minimise le score des adversaires
- maximise le minimum de la différence (mon score - celui de l'adversaire) sur les adversaires
- me donne la plus grande mobilité en m'enfermant le moins possible
- réduit la mobilité des adversaires en les isolant

## 2 | Algorithme minimax

On dresse le graphe des coups possibles jusqu'à la fin du jeu. Pour éviter d'explorer toutes les combinaisons possibles, il faut élaguer certaines branches, grâce au principe alpha-beta.

On se donne une autre contrainte pour limiter la complexité, parmi :

- un temps d'exécution maximal
- une profondeur d'exploration maximale

Pour améliorer cet algorithme, on peut aussi utiliser l'Upper Confidence on Trees (UCT) [se documenter] ou se focaliser sur les zones de points forts. Les zones de points forts correspondent à des zones qui permettent de s'assurer un score minimal. Par exemple si une situation nous permet de remporter 5 points dans le cas où l'adversaire joue d'une certaine façon, mais nous en fait perdre 10 si l'adversaire joue autrement, il faut éviter de se diriger vers cette situation.

### 2.1. Fonctions heuristiques

Se reporter aux différentes définitions du meilleur coup. On rajoute d'autres propositions :

- maximiser  $\sum_{i=2}^n sc - sc_i$
- utiliser une fonction d'utilité
- prendre en compte les zones d'influence des différents pions

La fonction d'utilité retourne, pour un état donné, la quantité :

$$U(s) = \alpha sc - \sum_{i=2}^n \beta_i sc_i + \gamma m - \sum_{i=2}^n \delta_i m_i$$

Il faut maximiser cette quantité. Pour déterminer les paramètres, on peut faire une régression linéaire sur les premiers coups et affiner les paramètres en cours de partie.

### 3 | Méthode de Monte-Carlo

Dans un état donné, on évalue la probabilité de gain sur chacun des états accessibles utiles (qui ne nous font pas perdre) et on choisit celui qui offre la plus grande probabilité de gain. Il faut ici aussi utiliser des méthodes d'utilisation pour forcer le hasard à ne pas se diriger vers des états inutiles.

### 4 | Réseau de neurones

On implémente un réseau de neurone capable de changer sa structure pour se spécialiser ou généraliser quand son comportement est mauvais.

On introduit un système de récompense pour favoriser l'évolution des paramètres du réseau dans le bon sens et ainsi donner un poids plus fort à des bons comportements.

On donne des exemples de base au réseau pour qu'il sache ce qu'il faut absolument faire dans une situation donnée, pour éviter de se faire bloquer, ou pour bloquer s'il le peut.