Université de Cergy-Pontoise,
UFR Sciences & Techniques,
Département des sciences informatiques,
Master 1 I&ISC

Intelligence Artificielle

TD 1 : Wargame

MiniMax, NegaMax et Alpha / Beta

P. Gaussier, L. Annabi & M. Abdelwahed

20 octobre 2020

L'objectif est de réaliser l'IA permettant de jouer (et de gagner aussi !) ce jeu de wargame. Pour cela, l'algorithme Minimax sera utilisé.

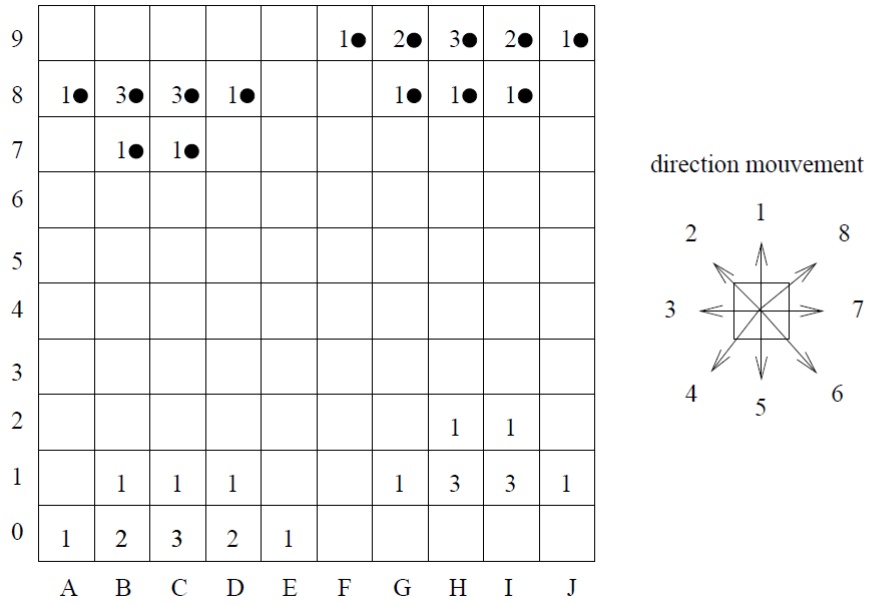

FIGURE 1 – Situation du jeu au départ et mouvements valides de chaque unité. Le chiffre indique la valeur d'une unité. Le point noir indique les unités appartenant aux Noirs. Les autres unités appartiennent aux Blancs. Ce sont les blancs qui commencent.

1 Cadre du jeu

Le but est de créer un "wargame" simplifié se jouant à 2 joueurs. A chaque tour, chaque joueur ne peut déplacer qu'une unité d'une case dans les 8 directions. Il est interdit de bouger vers une case occupée par une de ses unités ou de sortir de l'aire de jeu. Si la case destination est occupée par une unité ennemie, on prend l'unité ennemie à condition que la somme de ses pièces sur les 8 cases voisines soit supérieure à la somme des pièces ennemis sur ces même 8 cases voisines. Le perdant perd l'unité qui est mise en jeu. Un joueur perd s'il ne lui reste plus d'unité ou si l'autre joueur réussit à atteindre avec l'une de ses pièces, la ligne de fond de jeu de l'autre joueur.

Évaluation pour la prise d'une unité adverse sur la case (x,y) : B Gagnant ssi :

$$\sum_{i \in Vx,y} F_i^B > \sum_{i \in Vx,y} F_i^N$$

2 Introduction a l'algorithme Minimax

Une possibilité d'implémentation d'un programme permettant de jouer à ce jeu est l'utilisation de l'approche Minimax. Elle peut se résumer à l'exploration exhaustive de toutes les combinaisons de jeu. C'est l'approche la plus directe dans ce genre de jeu à état fini. Un exemple d'implémentation de l'algorithme est le suivant :

Algorithm 1: Pseudo code Minimax

```
function fmax(noeud, profondeur) is;
if profondeur maximale atteinte then
    return eval(noeud) ;
else
    value = -∞ ;
    for chaque fils(noeud) do
        value = max(value, fmin(fils, profondeur - 1));
    end
    return value;
end

function fmin(noeud, profondeur) is;
if profondeur maximale atteinte then
    return eval(noeud);
else
    value = +∞;
    for chaque fils(noeud) do
        value = min(value, fmax(fils, profondeur - 1));
    end
    return value;
end
```

L'algorithme fait une recherche en profondeur de toutes les possibilités de jeux possible en suivant les règles. La situation de jeu la plus favorable pour le joueur est la solution retenue par l'algorithme. Attention à bien implémenter la profondeur limite pour ne pas avoir une recherche qui prendrait un temps infini.

3 Fonction d'évaluation / heuristique

La fonction d'évaluation est la partie de l'algorithme qui permet d'évaluer une situation de jeu donnée. Elle doit prendre en considération les conditions de victoire mais pas uniquement. La manière de l'élaborer définira le comportement même de l'IA. Plusieurs paramètres peuvent être utilisés pour définir la fonction d'évaluation. A vous de trouvez ces paramètres.

4 Projet et rendu

Le rendu doit se composer du programme principal et d'un rapport.

Dans le programme principal, vous devez modifier les fonctions f_min, f_max, f_IA, et f_eval. Si votre code utilise d'autres fonctions ou structures que vous avez définies, merci de le noter explicitement dans votre rapport ou dans votre email pour faciliter l'interconnexion des IAs entre elles.

La profondeur de la recherche est fixée à 3 pour le tournois.

L'IA implentera le Minimax ainsi que les coupures alpha / beta. Le negamax peut être utilisé à la place du minimax.

Un rapport évaluant les performances de votre algorithme, de 2 à 5 pages, est attendu. Il devra comporter une partie sur l'analyse des coupures alpha-beta. On pourra par exemple coder un compteur qui comptera le nombre de noeuds visités lors de l'exploration de l'arbre, ou mesurer le temps de calcul lors de l'appel de la fonction f_IA. Il devra également justifier vos choix d'implémentation de la fonction d'évaluation, et analyser leurs effets sur les performances de votre IA.