

Implémentation d'un algorithme Alpha/Béta ou SSS* et Application au jeu de Chomp

Note : Ce sujet a été donné lors de ma formation universitaire à l'[UPS](#) (Université Paul Sabatier) en cours d'intelligence artificielle.

1. Algorithme de calcul de la valeur Négamax.
2. Algorithme de calcul de la valeur Négamax en utilisant la borne supérieure de l'intervalle d'utilité.
3. Algorithme de calcul de la valeur Négamax en utilisant la borne inférieure et supérieure de l'intervalle d'utilité.
4. Les règles du jeu de Chomp.
5. Définition d'une heuristique : méthodologie.

1. Algorithme de calcul de la valeur Négamax

Fonction : **estimation1**

Arguments :

noeud (* celui que l'on évalue *)

Variables globales :

mp (* le maximum provisoire *)

n (* le nombre de fils du noeud à traiter *)

i (* un compteur de boucle *)

vloc (* pour sauvegarder des résultats *)

Valeur retournée :

La valeur **négamax** du noeud argument

Algorithme :

début

si noeud est terminal

alors estimation1 <- estimation initiale de cette feuille

sinon

début

développer **noeud**; (* soit **n** le nombre de ses fils *)

i <- 1

mp <- $-\infty$;

tant_que $i \leq n$ **faire**

début

vloc <- - **estimation1**(**fi**(**noeud**));

si vloc > **mp**

alors mp <- **vloc**;

i <- **i** + 1

fin

estimation1 <- **mp**

fin

fin

Le calcul de la valeur négamax d'un noeud **n** s'obtient en appelant **estimation1**(**n**).

2. Algorithme de calcul de la valeur Négamax, en utilisant la borne supérieure de l'intervalle d'utilité

Fonction : **estimation2**

Arguments :

noeud (* celui que l'on évalue *)

b (* borne supérieure de l'intervalle d'utilité de **noeud** *)

Variables globales :

mp (* le maximum provisoire *)

n (* le nombre de fils du noeud à traiter *)

i (* un compteur de boucle *)

vloc (* pour sauvegarder des résultats *)

Valeur retournée :

La valeur **négamax** du noeud argument

Algorithme :

début

si **noeud** est terminal

alors **estimation2** <- estimation initiale de cette feuille

sinon

début

développer **noeud**; (* soit **n** le nombre de ses fils *)

i <- 1

mp <- $-\infty$;

tant que $i \leq n$ et **mp** < **b** faire

début

vloc <- - **estimation2**(**fi**(**noeud**), - **mp**);

si **vloc** > **mp**

alors **mp** <- **vloc**;

i <- **i** + 1

fin

estimation2 <- **mp**

fin

fin

La valeur initiale de b sera l' ∞ .

Le calcul de la valeur négamax d'un noeud **n** s'obtient en appelant **estimation2**(**n**, $+\infty$).

3. Algorithme de calcul de la valeur Négamax, en utilisant la borne inférieure et la borne supérieure de l'intervalle d'utilité

Fonction : **estimation3**

Arguments :

noeud (* celui que l'on évalue *)

mp (* borne inférieure de l'intervalle d'utilité et valeur courante de **noeud** *)

b (* borne supérieure de l'intervalle d'utilité de **noeud** *)

Variables globales :

n (* le nombre de fils du noeud à traiter *)

i (* un compteur de boucle *)

vloc (* pour sauvegarder des résultats *)

Valeur retournée :

La valeur **négamax** du noeud argument

Algorithme :

début

si **noeud** est terminal

alors **estimation3** <- estimation initiale de cette feuille

sinon

début

développer **noeud**; (* soit **n** le nombre de ses fils *)

i <- 1

tant_que **i** ≤ **n** et **mp** < **b** faire

début

vloc <- - **estimation3**(**fi**(**noeud**), -**b**, - **mp**);

si **vloc** > **mp**

alors **mp** <- **vloc**;

i <- **i** + 1

fin

estimation3 <- **mp**

fin

fin

Le calcul de la valeur négamax d'un noeud **n** s'obtient en appelant **estimation3**(**n**, $-\infty$, $+\infty$).

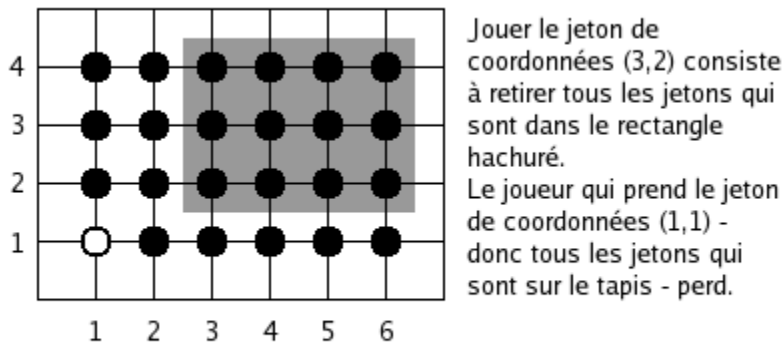
4. Les règles du jeu de Chomp

4.1 Description du jeu.

a) Le jeu se joue sur un tapis comportant un repère orthonormé. Sur ce tapis sont disposés tous les jetons de coordonnées (u, v) avec $1 \leq u \leq m$ et $1 \leq v \leq n$ où m et n sont supérieurs ou égaux à 1 et le produit $m*n$ est strictement supérieur à 1.

Un jeu est donc caractérisé par le couple (m, n) . En conséquence, on parlera d'un jeu de type $m*n$.

- b) Les adversaires seront dénotés respectivement par A et B. Ils jouent à tour de rôle, A joue le premier.
- c) Jouer c'est choisir un jeton de coordonnées (x,y) sur le tapis et prélever tous les jetons de coordonnées (x', y') situés sur le tapis tels que $x' \geq x$ et $y' \geq y$.
- d) C'est le joueur qui par son coup prélève le jeton de coordonnées $(1, 1)$ qui perd.



Exemple de jeu $6 * 4$

4.2 Jeu programmé.

Dans le jeu programmé, la machine pourra indiquer les coordonnées du jeton qu'elle joue. C'est le joueur qui reproduira alors l'état du tapis sur une feuille de papier.

La machine devra utiliser un algorithme de recherche, l'algorithme Alpha-Béta, paramétrable par la profondeur de l'arbre de recherche, et une "bonne" fonction d'estimation.

5. Définition d'une heuristique : méthodologie

On peut démontrer qu'il existe une stratégie gagnante pour le premier qui joue dans le cas de tout jeu de type $m*n$ (avec $m*n > 1$).

Cette stratégie gagnante est aisée à décrire dans les cas suivants :
jeu de type $1*n$ (ou $m*1$), de type $2*n$ (ou $m*2$), de type $m*m$.