

# Intelligence artificielle

ENSIIE

Semestre 4 – 2020/21

# Algorithmes de jeux

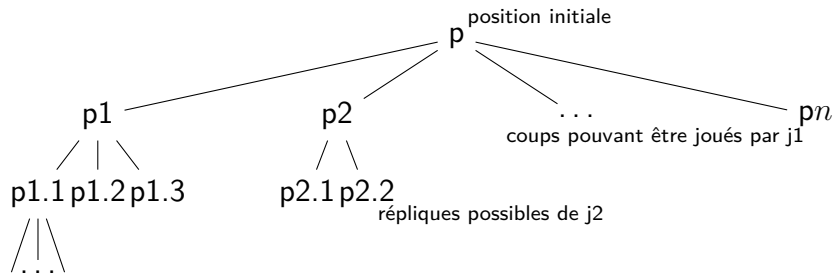
# Jeux

Dans ce cours, essentiellement :

- ▶ jeux à deux joueurs
- ▶ séquentiels
- ▶ symétriques
- ▶ à somme nulle (si l'un gagne, l'autre perd)
- ▶ chaque adversaire a une connaissance de l'ensemble du jeu et de ce que chacun peut faire
- ▶ sans aléa

# Arbre et/ou

Les différentes parties possibles peuvent être représentées par un arbre



Feuilles : état terminal dans lequel  $j_1$  a gagné, fait nul ou perdu.

## Procédé minimax

Construction de l'arbre, ou d'une partie

- ▶ Différents coups jusqu'à une profondeur limitée

Évaluation des feuilles

- ▶ Fonction d'évaluation qu'on cherche à maximiser
- ▶ Évalue une situation pour j1 (la machine)

Remontée des valeurs : principe min ou max selon le joueur

- ▶ Une position où j1 joue = la valeur max de ses fils
- ▶ Une position où j2 joue = la valeur min de ses fils

(Suppose une stratégie identique pour les 2 joueurs)

## Exemple de fonction d'évaluation

Au morpion :

- ▶ arbre complet
- ▶ gagné = 1, perdu = -1, nul = 0

Aux échecs :

- ▶ arbre incomplet !
- ▶ 
$$\sum_{\text{type de pièce } t} w_t \times d_t(a)$$
avec  $w_t$  un poids pour les pièces de type  $t$   
 $d_t(a)$  différence entre le nombre de pièces du type  $t$  dans la configuration  $a$

## Caractéristiques de minimax

- ▶ Complète si l'arbre est fini
- ▶ Complexité en temps :  $O(b^m)$
- ▶ Complexité en espace :  $O(bm)$
- ▶ Optimal si l'adversaire est optimal

## Efficacité de minimax

Aux échecs,  $b \simeq 35$

Dans un temps donné, si on veut explorer  $b^m = 10^6$  nœuds, on peut donc aller au plus à la profondeur  $m = 4$

Idée : ne pas calculer de nœuds inutilement

- ▶ dont on sait déjà qu'ils seront inférieurs au max ou supérieur au min



## Élagage $\alpha\beta$

Intuition :

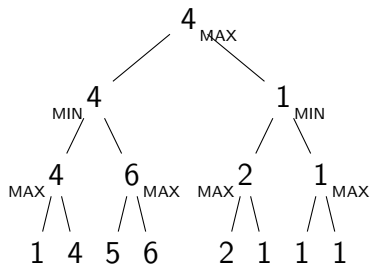
On parcourt les nœuds avec une fenêtre  $\alpha \dots \beta$

- ▶ on est sûr que MAX renverra au moins  $\alpha$
- ▶ on est sûr que MIN renverra au plus  $\beta$

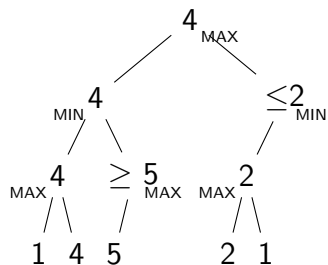
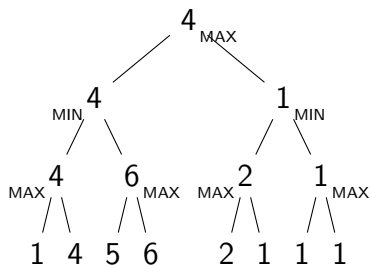
Le calcul de la valeur d'un nœud permet d'affiner la fenêtre pour le calcul de ses frères

Pas besoin de calculer si fenêtre vide

## Exemple



## Exemple



Algorithme  $\alpha\beta$ 

```

fonction alphabeta(nœud,  $\alpha$ ,  $\beta$ )
  si nœud est une feuille alors retourner la valeur de nœud
  sinon si nœud est de type Min alors
     $v = +\infty$ 
    pour tout fils de nœud faire
       $v = \min(v, \text{alphabeta}(\text{fils}, \alpha, \beta))$ 
      si  $v \leq \alpha$  alors /* coupure alpha */
        retourner  $v$ 
       $\beta = \text{Min}(\beta, v)$ 
  sinon
     $v = -\infty$ 
    pour tout fils de nœud faire
       $v = \max(v, \text{alphabeta}(\text{fils}, \alpha, \beta))$ 
      si  $v \geq \beta$  alors /* coupure beta */
        retourner  $v$ 
       $\alpha = \text{Max}(\alpha, v)$ 
  retourner  $v$ 

```