

Intelligence Artificielle (IA)

Les jeux, recherche avec horizon (II)

Akka Zemmari

LaBRI, Université de Bordeaux

2023 - 2024

$\alpha\beta$

Élagage efficace de l'arbre

Élagage admissible

On veut trouver la même valeur d'évaluation finale du noeud racine sans développer tout l'arbre. Il faut donc élaguer des parties de l'arbre de recherche qui sont sans conséquence sur l'évaluation d'un noeud.

$\alpha\beta$

Élagage efficace de l'arbre

Élagage admissible

On veut trouver la même valeur d'évaluation finale du noeud racine sans développer tout l'arbre. Il faut donc élaguer des parties de l'arbre de recherche qui sont sans conséquence sur l'évaluation d'un noeud.

Intuitivement

Soit un noeud n dans l'arbre de recherche, tel que *Joueur* peut jouer en n .

S'il existe pour *Joueur* un choix m meilleur que n (soit à partir du noeud parent de n , soit plus haut dans l'arbre), n ne sera jamais effectivement joué.

$\alpha\beta$

Elagage efficace de l'arbre II

Deux types de coupes : α et β

- ▶ α : le meilleur choix à un instant donné pour Max sur le chemin développé. **La valeur α est croissante**
- ▶ β : le meilleur choix à un instant donné pour Min sur le chemin développé. **La valeur β est décroissante**

Les coupes auront lieu dès que α est supérieur à β

Algorithme $\alpha\beta$

La première fonction : *MaxValue*

- 1: **Fonction** *MaxValue*(*etat*, α , β) ▷ Évaluation niveau AMI
- 2: *etat* : Plateau de jeu courant
- 3: α : Meilleure évaluation courante pour AMI
- 4: β : Meilleure évaluation courante pour ENNEMI
- 5: **Si** *EstFeuille*(*etat*) **Alors**
- 6: **Retourner** *evaluate*(*etat*) ▷ Évaluation heuristique
- 7: **Fin Si**
- 8: **Pour Tout** successeur *s* de *etat* **Faire**
- 9: $\alpha \leftarrow \max(\alpha, \text{MinValue}(s, \alpha, \beta))$
- 10: **Si** $\alpha \geq \beta$ **Alors** ▷ Coupe β
- 11: **Retourner** β
- 12: **Fin Si**
- 13: **Fin Pour**
- 14: **Retourner** α
- 15: **Fin Fonction**

Algorithme $\alpha\beta$

La suite : *MinValue*

Remarque. - Pour évaluer un plateau, on appelle *MaxValue* avec :
plateau à évaluer, $\alpha = -\infty$ et $\beta = +\infty$. Les variables α et β sont bien
locales, elles sont changées par l'intermédiaire des valeurs de retour.

```

1: Fonction MinValue(etat,  $\alpha$ ,  $\beta$ )
2:   etat : Plateau de jeu courant
3:    $\alpha$  : Meilleure évaluation courante pour AMI
4:    $\beta$  : Meilleure évaluation courante pour ENNEMI
5:   Si EstFeuille(etat) Alors
6:     Retourner evaluer(etat)
7:   Fin Si
8:   Pour Tout successeur s de etat Faire
9:      $\beta \leftarrow \min(\beta, \text{MaxValue}(s, \alpha, \beta))$ 
10:    Si  $\alpha \geq \beta$  Alors
11:      Retourner  $\alpha$ 
12:    Fin Si
13:  Fin Pour
14:  Retourner  $\beta$ 
15: Fin Fonction

```

▷ Évaluation niveau ENNEMI
 ▷ Évaluation heuristique
 ▷ Coupe α

Algorithme $\alpha\beta$, Les deux fonctions ensemble

Fonction *MaxValue*(*etat*, α , β)

Si *EstFeuille*(*etat*) **Alors**

Retourner *evaluate*(*etat*)

Fin Si

Pour Tout successeur *s* de *etat* **Faire**

$\alpha \leftarrow \max(\alpha, \text{MinValue}(s, \alpha, \beta))$

Si $\alpha \geq \beta$ **Alors**

Retourner β

Fin Si

Fin Pour

Retourner α

Fin Fonction

Fonction *MinValue*(*etat*, α , β)

Si *EstFeuille*(*etat*) **Alors**

Retourner *evaluate*(*etat*)

Fin Si

Pour Tout successeur *s* de *etat* **Faire**

$\beta \leftarrow \min(\beta, \text{MaxValue}(s, \alpha, \beta))$

Si $\alpha \geq \beta$ **Alors**

Retourner α

Fin Si

Fin Pour

Retourner β

Fin Fonction

▷ Niveau AMI

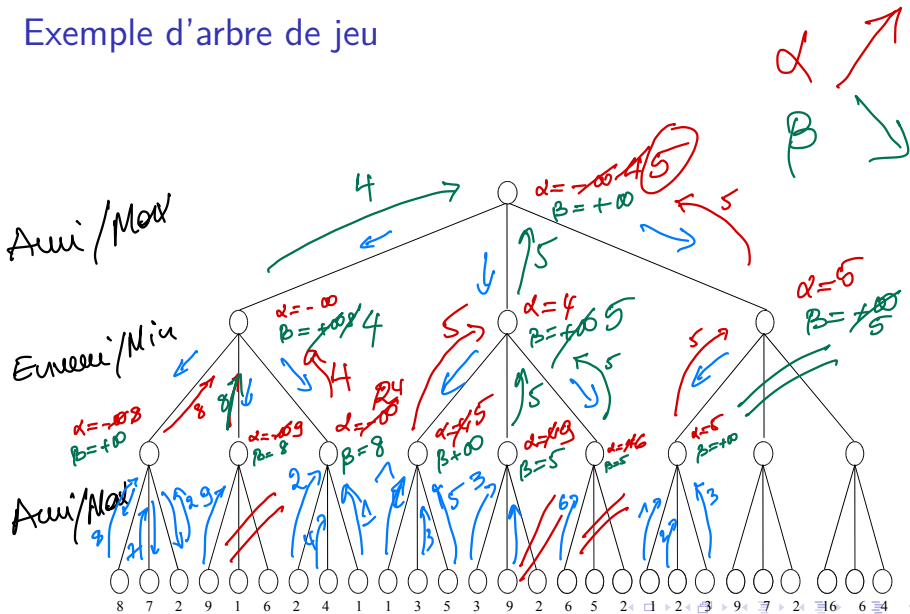
▷ Évaluation heuristique

▷ Coupe β

▷ Niveau ENNEMI

▷ Coupe α

Exemple d'arbre de jeu



Propriétés de $\alpha\beta$

Efficacité théorique

Si on suppose que les fils sont ordonnés idéalement (ou presque) :

- ▶ La recherche peut aller deux fois plus loin dans l'arbre

Améliorations possibles

- ▶ Comment couper encore plus lors de la recherche ?
 - ▶ Recherche aspirante : α et β sont initialisés.
 - ▶ Si la valeur finale est bien dans $[\alpha, \beta]$ la recherche reste admissible
- ▶ ... D'autres encore à venir !