Université de Bourgogne
1ère année de Master Informatique STIC
Parcours Informatique
Algorithme et complexité

Le Jeu Othello

Ghislain Loaëc - Abdeldjalil Ramoul
2012-2013

Plan

- Introduction
- Le jeu Othello
- Algorithme du Minimax
- Elagage *Alpha-beta*
- *Démonstration de l'application*
- *Conclusion et Perspectives*

Plan

- Introduction
- Le jeu Othello
- Algorithme du Minimax
- Elagage *Alpha-beta*
- *Démonstration de l'application*
- *Conclusion et Perspectives*

Introduction

- développement d’un simulateur du jeu Othello
- Présentation de Othello
- Utilisation de la méthode Minimax pour jouer à Othello

Plan

- Introduction
- Le jeu Othello
- Algorithme du Minimax
- Elagage *Alpha-beta*
- *Démonstration de l'application*
- *Conclusion et Perspectives*

Le jeu Othello

Le jeu Othello

- jeu de société à deux joueurs à information parfaite avec un nombre fini de stratégies
- absence de l'incertitude => existence d'une solution optimale
- Utilisation des heuristiques pour donner une approximation du jeu optimal

Règles du jeu

- Othellier unicolore de 64 cases
- Début de partie: blancs =>D4, E5 | noirs =>E4 et D5.
- Capturer les jetons de l’adversaire
- A la fin de la partie compter les jetons

Plan

- Introduction
- Le jeu Othello
- Algorithme du Minimax
- Elagage *Alpha-beta*
- *Démonstration de l'application*
- *Conclusion et Perspectives*

Algorithme du minimax

- Basé sur le théorème du même nom (1928)
- Utilisé pour la découverte de la solution optimale dans un jeu à deux joueurs à information parfaite
- Principe: visiter l'arbre récursivement et faire remonter la valeur minimax à la racine.

Algorithme du minimax

- *Minimax(e) = h(e), si e est une feuille de l'arbre*
- *Minimax(e) = max (Minimax(e1), . . . , Minimax(en)), si e est un nœud Joueur*
- *Minimax(e) = min (Minimax(e1), . . . , Minimax(en)), si e est un nœud Adversaire*

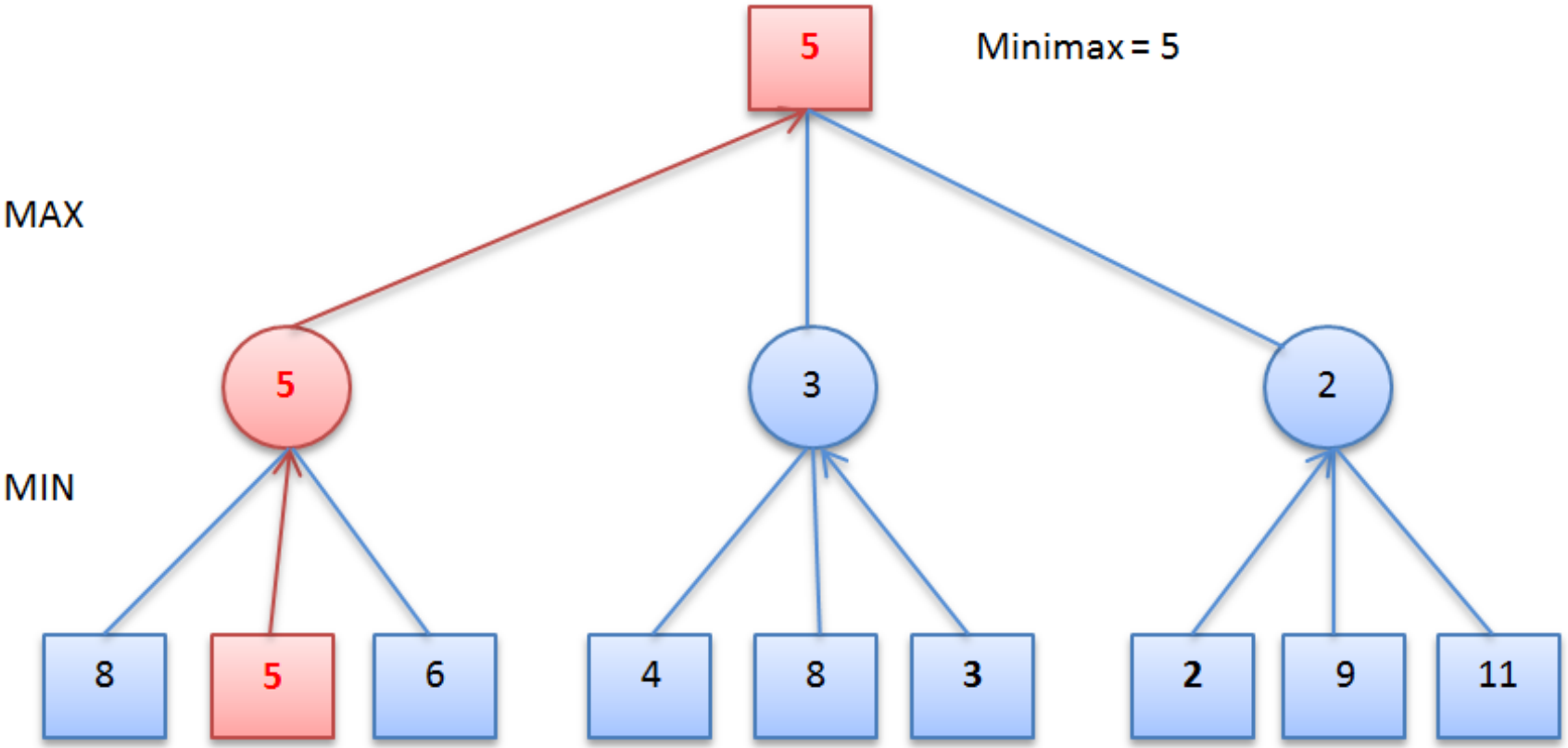
Algorithme du minimax

complexité en temps = O(b^m)
complexité en espace = O(bm)
b = facteur de branchement
m = est la profondeur de l'arbre

Fonction Minimax (e,**d**)

Entrées : nœud *e*, **profondeur *d***
Sorties : Valeur Minimax du nœud *e*
Si *final ?(e) ou* ***(d == 0)*** **alors**
    Return *h*(*e*)
Sinon
    Si *joueur ?(e)* **alors**
        Return *max*{*Minimax*(*ei* , ***d* – 1**) | *ei* > *f*(*e*)}
    Sinon
        Return *min*{*Minimax*(*ei* , ***d* – 1**) | *ei* > *f*(*e*)}

Algorithme du minimax



Algorithme du minimax

- Problème d'explosion combinatoire
- Profondeur limitée => pas très performant
- Elagage alpha-beta => augmenter la profondeur
- Réduire la taille de l'arbre de jeu en élaguant les parties dont l'évaluation ne contribue pas à celle de la racine

Plan

- Introduction
- Le jeu Othello
- Algorithme du Minimax
- Elagage *Alpha-beta*
- *Démonstration de l'application*
- *Conclusion et Perspectives*

Elagage alpha-beta

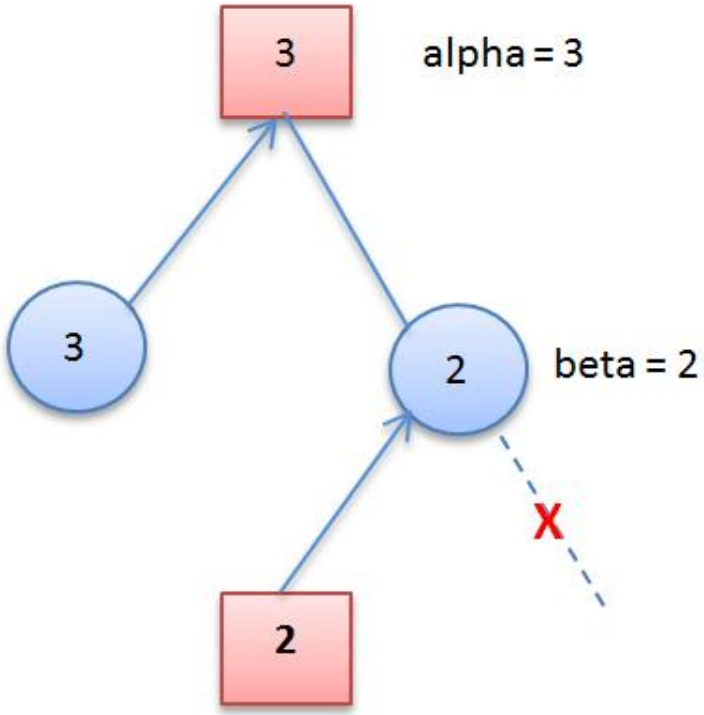
Alpha représente de la borne inférieure de la valeur du nœud :

- Alpha = h(e) sur les feuilles, et initialisée à -1 ailleurs.
- Dans les nœuds joueurs, alpha = MAX des valeurs obtenues sur les fils visités jusque-là.
- Dans les nœuds adversaires, elle est égale à la valeur de son prédécesseur.

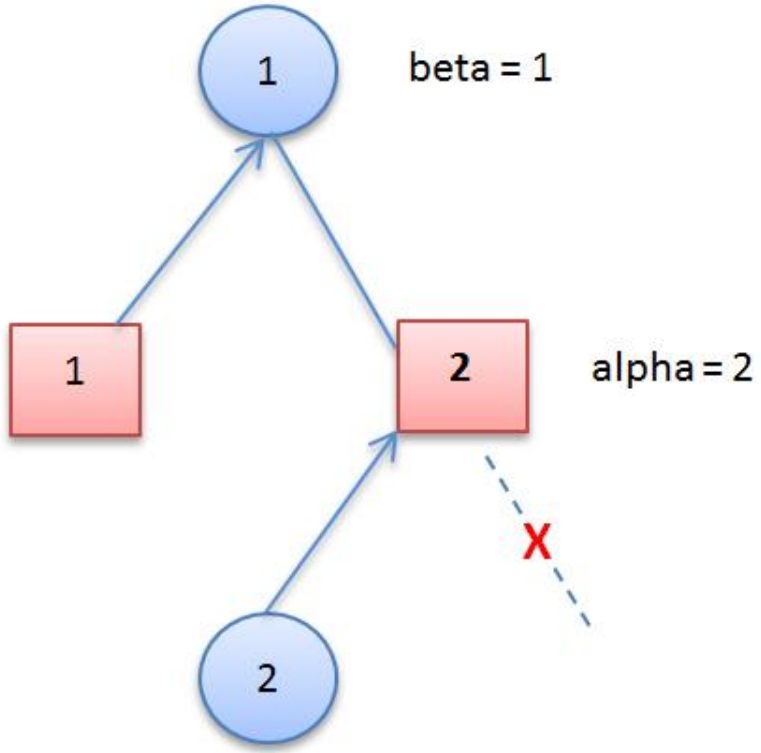
Beta représente de la borne supérieure de la valeur du nœud :

- Beta = h(e) sur les feuilles, et initialisée à +1 ailleurs.
- Dans les nœuds adversaires, beta = MIN des valeurs obtenues sur les fils visités jusque-là.
- Dans les nœuds joueurs elle est égale à la valeur de son prédécesseur.

Elagage alpha-beta



Alpha-coupure



Beta-coupure

Elagage alpha-beta

Algorithme 3

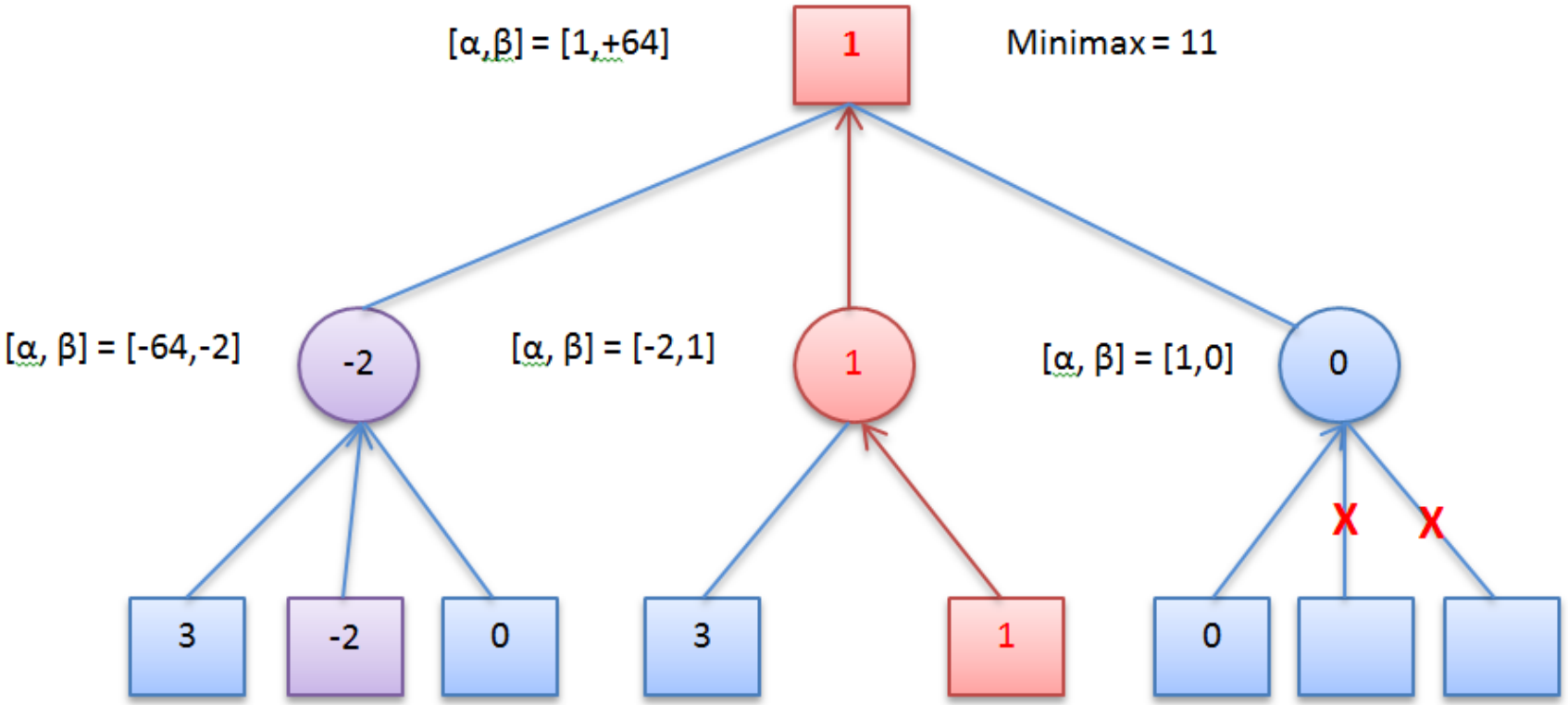
Fonction Alphabeta(e, d, α, β)

Entrées: noeud e, profondeur d
Sorties : valeur Minimax du noeud e

```
Debut
    Si final ? (e) ou (d == 0) alors
      Return h(e)
    Sinon
        Si joueur ? (e) alors
          v = -64
          pour fi > f(e)faire
              v = max(v, Alphabeta(fi, d - 1, α , β))
              Si v > β alors return v
              a = max(a, v)
          Finpour
        Sinon
            v = +64
            Pour tous les fi ∈ f(e) faire
                v = min(v, Alphabeta(fi, d -1, α, β))
                Si α > v alors return v
                β = min(β, v)
            Finpour
         Finsi
      Finsi
      return v
Fin
```

Complexité dans les meilleurs des cas O(b^(m/2))

Elagage alpha-beta



Plan

- Introduction
- Le jeu Othello
- Algorithme du Minimax
- Elagage *Alpha-beta*
- *Démonstration de l'application*
- *Conclusion et Perspectives*

Plan

- Introduction
- Le jeu Othello
- Algorithme du Minimax
- Elagage *Alpha-beta*
- *Démonstration de l'application*
- *Conclusion et Perspectives*

Conclusion et Perspectives

- Jeu simple mais tactiques nombreuses
- Fonction heuristique simple:

  H(e) = Jetons joueur – jetons adversaire

- Amélioration de H(e) avec des valeurs tactique

500	-150	30	10	10	30	-150	500
-150	-250	0	0	0	0	-250	-150
30	0	1	2	2	1	0	30
10	0	2	16	16	2	0	10
10	0	2	16	16	2	0	10
30	0	1	2	2	1	0	30
-150	-250	0	0	0	0	-250	-150
500	-150	30	10	10	30	-150	500