

# Intelligence Artificielle (IA)

## Les jeux, recherche avec horizon (III)

### A la recherche de la performance

Akka Zemmari

LaBRI, Université de Bordeaux

2021 - 2022

## Problématique du temps réel

### Questions liées au choix de la profondeur de recherche

**La profondeur de recherche doit être fixée avant de lancer la recherche** (L'horizon doit être équilibré sur tout l'arbre de recherche). Comment choisir la profondeur ?

- ▶ Comment garantir au joueur que l'ordinateur ne va pas passer trop de temps à réfléchir ?

## Problématique du temps réel

### Questions liées au choix de la profondeur de recherche

**La profondeur de recherche doit être fixée avant de lancer la recherche** (L'horizon doit être équilibré sur tout l'arbre de recherche). Comment choisir la profondeur ?

- ▶ Comment garantir au joueur que l'ordinateur ne va pas passer trop de temps à réfléchir ?
- ▶ Comment le garantir avec des machines différentes ?

## Problématique du temps réel

### Questions liées au choix de la profondeur de recherche

**La profondeur de recherche doit être fixée avant de lancer la recherche** (L'horizon doit être équilibré sur tout l'arbre de recherche). Comment choisir la profondeur ?

- ▶ Comment garantir au joueur que l'ordinateur ne va pas passer trop de temps à réfléchir ?
- ▶ Comment le garantir avec des machines différentes ?
- ▶ Comment lier le temps passé à réfléchir avec la profondeur maximale de l'horizon ?

## Problématique du temps réel

### Questions liées au choix de la profondeur de recherche

**La profondeur de recherche doit être fixée avant de lancer la recherche** (L'horizon doit être équilibré sur tout l'arbre de recherche). Comment choisir la profondeur ?

- ▶ Comment garantir au joueur que l'ordinateur ne va pas passer trop de temps à réfléchir ?
- ▶ Comment le garantir avec des machines différentes ?
- ▶ Comment lier le temps passé à réfléchir avec la profondeur maximale de l'horizon ?
- ▶ Que se passe-t-il si l'arbre n'est pas homogène (hypothèse (très) réaliste) ?

## *Iterative Deepening*

Maîtriser le temps

Idée : Étendre peu à peu l'horizon de  $\alpha\beta$

## *Iterative Deepening*

Maîtriser le temps

Idée : Étendre peu à peu l'horizon de  $\alpha\beta$

Mais doit-on mémoriser tout l'arbre pour repartir des feuilles ?

## Iterative Deepening

### Maîtriser le temps

Idée : Étendre peu à peu l'horizon de  $\alpha\beta$

Mais doit-on mémoriser tout l'arbre pour repartir des feuilles ?

### Iterative Deepening (ID)

Soit  $p$  initialisé à un horizon *immédiat*.

- ▶ Faire une recherche à horizon  $p$
- ▶ S'il reste du temps, tout oublier (sauf le coup à jouer) et incrémenter  $p$ .



## Iterative Deepening

### Maîtriser le temps

Idée : Étendre peu à peu l'horizon de  $\alpha\beta$

Mais doit-on mémoriser tout l'arbre pour repartir des feuilles ?

### Iterative Deepening (ID)

Soit  $p$  initialisé à un horizon *immédiat*.

- ▶ Faire une recherche à horizon  $p$
- ▶ S'il reste du temps, tout oublier (sauf le coup à jouer) et incrémenter  $p$ .

## Iterative Deepening

### Maîtriser le temps

**Idee :** Étendre peu à peu l'horizon de  $\alpha\beta$

Mais doit-on mémoriser tout l'arbre pour repartir des feuilles ?

### Iterative Deepening (ID)

Soit  $p$  initialisé à un horizon *immédiat*.

- ▶ Faire une recherche à horizon  $p$
- ▶ S'il reste du temps, tout oublier (sauf le coup à jouer) et incrémenter  $p$ .

**Et les performances ?**

ID semble refaire beaucoup de fois la même chose.

## Iterative Deepening

### Maîtriser le temps

Idee : Étendre peu à peu l'horizon de  $\alpha\beta$

Mais doit-on mémoriser tout l'arbre pour repartir des feuilles ?

### Iterative Deepening (ID)

Soit  $p$  initialisé à un horizon *immédiat*.

- ▶ Faire une recherche à horizon  $p$
- ▶ S'il reste du temps, tout oublier (sauf le coup à jouer) et incrémenter  $p$ .

Et les performances ?

ID semble refaire beaucoup de fois la même chose. Est-ce pour autant **très inefficace** ?

## *Iterative Deepening*

Un algorithme pas si *inefficace*

La croissance exponentielle des arbres montre intuitivement que la majorité des noeuds (donc de la difficulté) se situent sur les feuilles...

## *Iterative Deepening*

Un algorithme pas si *inefficace*

La croissance exponentielle des arbres montre intuitivement que la majorité des noeuds (donc de la difficulté) se situent sur les feuilles... **Développer le dernier niveau coûte le plus cher**, d'autant plus si on a un grand facteur de branchement.

## Iterative Deepening

Un algorithme pas si *inefficace*

La croissance exponentielle des arbres montre intuitivement que la majorité des noeuds (donc de la difficulté) se situent sur les feuilles... **Développer le dernier niveau coûte le plus cher**, d'autant plus si on a un grand facteur de branchement.

Qui veut des chiffres ?

Une recherche à profondeur  $p$  et facteur de branchement  $b$  coûte

$$1 + b + b^2 + \dots + b^{p-1} + b^p$$

Pour ( $p = 5, b = 10$ ) on a 11 111 noeuds.

## Iterative Deepening

Un algorithme pas si *inefficace*

La croissance exponentielle des arbres montre intuitivement que la majorité des noeuds (donc de la difficulté) se situent sur les feuilles... **Développer le dernier niveau coûte le plus cher**, d'autant plus si on a un grand facteur de branchement.

Qui veut des chiffres ?

Une recherche à profondeur  $p$  et facteur de branchement  $b$  coûte

$$1 + b + b^2 + \dots + b^{p-1} + b^p$$

Pour ( $p = 5, b = 10$ ) on a 11 111 noeuds. Si on fait un ID, on aurait développé 12 345 noeuds au total (1 pour  $d=1$ , 11 pour  $d=2$ , 111 pour  $d=3$ , 1 111 pour  $d=4$  ...), soit **11% de noeuds supplémentaires** (seulement).

## *Iterative Deepening*

Des avantages de poids avec  $\alpha\beta$

### Avantages

- ▶ Grande souplesse dans la gestion du temps. Le programme peut donner la meilleur valeur trouvée à n'importe quel moment.
- ▶ Couplé à  $\alpha\beta$ , ID peut même devenir plus efficace qu'un seul  $\alpha\beta$



## Iterative Deepening

Des avantages de poids avec  $\alpha\beta$

### Avantages

- ▶ Grande souplesse dans la gestion du temps. Le programme peut donner la meilleur valeur trouvée à n'importe quel moment.
- ▶ Couplé à  $\alpha\beta$ , ID peut même devenir plus efficace qu'un seul  $\alpha\beta$

### Comment ?

$\alpha\beta$  élague d'autant plus que les meilleurs coups sont développés en premier. On profite donc de l'ancien  $\alpha\beta$  à profondeur  $p$  pour ordonner les fils à profondeur  $p + 1$ .

## *Iterative Deepening*

En pratique

### Idées

Plutôt que de tout oublier après chaque relance de ID, on va essayer de mémoriser pour réordonner l'arbre ensuite.

## *Iterative Deepening*

En pratique

### Idées

Plutôt que de tout oublier après chaque relance de ID, on va essayer de mémoriser pour réordonner l'arbre ensuite.

### Première méthode

Garder tout l'arbre de recherche en mémoire,

## Iterative Deepening

En pratique

### Idées

Plutôt que de tout oublier après chaque relance de ID, on va essayer de mémoriser pour réordonner l'arbre ensuite.

### Première méthode

Garder tout l'arbre de recherche en mémoire, l'ordonner à chaque niveau puis relancer  $\alpha\beta$  directement sur le nouvel arbre *optimal*.

## Iterative Deepening

### En pratique

#### Idées

Plutôt que de tout oublier après chaque relance de ID, on va essayer de mémoriser pour réordonner l'arbre ensuite.

#### Première méthode

Garder tout l'arbre de recherche en mémoire, l'ordonner à chaque niveau puis relancer  $\alpha\beta$  directement sur le nouvel arbre *optimal*.

#### Seconde méthode

Réserver une zone mémoire pour associer à chaque plateau de jeu déjà vu le meilleur des fils et la valeur heuristique associée (ainsi que sa profondeur).

## Problèmes à l'horizon

### Inconvénients de fixer un horizon

Tous les programmes doivent fixer un horizon. Pourtant, cela entraîne deux problèmes importants :

- ▶ **Manque de clairvoyance** : On ne peut prévoir un coup (même fatal) au-delà de l'horizon

## Problèmes à l'horizon

### Inconvénients de fixer un horizon

Tous les programmes doivent fixer un horizon. Pourtant, cela entraîne deux problèmes importants :

- ▶ **Manque de clairvoyance** : On ne peut prévoir un coup (même fatal) au-delà de l'horizon
- ▶ **Aveuglement volontaire** : l'approche MiniMax va pousser le programme à tout faire pour repousser un coup mauvais – mais inévitable – au delà de l'horizon.

## Problèmes à l'horizon

### Inconvénients de fixer un horizon

Tous les programmes doivent fixer un horizon. Pourtant, cela entraîne deux problèmes importants :

- ▶ **Manque de clairvoyance** : On ne peut prévoir un coup (même fatal) au-delà de l'horizon
- ▶ **Aveuglement volontaire** : l'approche MiniMax va pousser le programme à tout faire pour repousser un coup mauvais – mais inévitable – au delà de l'horizon.

Deux solutions... Parmi d'autres



## Problèmes à l'horizon

### Inconvénients de fixer un horizon

Tous les programmes doivent fixer un horizon. Pourtant, cela entraîne deux problèmes importants :

- ▶ **Manque de clairvoyance** : On ne peut prévoir un coup (même fatal) au-delà de l'horizon
- ▶ **Aveuglement volontaire** : l'approche MiniMax va pousser le programme à tout faire pour repousser un coup mauvais – mais inévitable – au delà de l'horizon.

### Deux solutions... Parmi d'autres

- ▶ **Déséquilibrer l'arbre de recherche**

## Problèmes à l'horizon

### Inconvénients de fixer un horizon

Tous les programmes doivent fixer un horizon. Pourtant, cela entraîne deux problèmes importants :

- ▶ **Manque de clairvoyance** : On ne peut prévoir un coup (même fatal) au-delà de l'horizon
- ▶ **Aveuglement volontaire** : l'approche MiniMax va pousser le programme à tout faire pour repousser un coup mauvais – mais inévitable – au delà de l'horizon.

### Deux solutions... Parmi d'autres

- ▶ Déséquilibrer l'arbre de recherche
- ▶ Utiliser une *méta-heuristique*

# Atténuation d'horizons

Zoomer où l'on va aller

## Idée

Une fois que l'on a fait le choix du coup à jouer, passer un peu de temps pour voir si la fonction heuristique ne nous a pas trompé.

## Atténuation d'horizons

Zoomer où l'on va aller

### Idée

Une fois que l'on a fait le choix du coup à jouer, passer un peu de temps pour voir si la fonction heuristique ne nous a pas trompé.  
On va déséquilibrer l'arbre de recherche.

Deux solutions... Parmi d'autres

- ▶ Redévelopper un  $\alpha\beta$  sur la branche sélectionnée.

## Atténuation d'horizons

Zoomer où l'on va aller

### Idée

Une fois que l'on a fait le choix du coup à jouer, passer un peu de temps pour voir si la fonction heuristique ne nous a pas trompé.  
On va déséquilibrer l'arbre de recherche.

Deux solutions... Parmi d'autres

- ▶ Redévelopper un  $\alpha\beta$  sur la branche sélectionnée.

## Atténuation d'horizons

Zoomer où l'on va aller

### Idée

Une fois que l'on a fait le choix du coup à jouer, passer un peu de temps pour voir si la fonction heuristique ne nous a pas trompé.  
On va déséquilibrer l'arbre de recherche.

### Deux solutions... Parmi d'autres

- ▶ Redévelopper un  $\alpha\beta$  sur la branche sélectionnée.
- ▶ Redévelopper un  $\alpha\beta$  sur la meilleure feuille visée.

## Atténuation d'horizons

Zoomer où l'on va aller

### Idée

Une fois que l'on a fait le choix du coup à jouer, passer un peu de temps pour voir si la fonction heuristique ne nous a pas trompé.  
On va déséquilibrer l'arbre de recherche.

### Deux solutions... Parmi d'autres

- ▶ Redévelopper un  $\alpha\beta$  sur la branche sélectionnée.
- ▶ Redévelopper un  $\alpha\beta$  sur la meilleure feuille visée.

# Atténuation d'horizons

Reconsidérer la notion d'horizons

## Idée

Définir l'horizon en fonction de l'intérêt des coups joués et non en fonction du nombre de coups.



## Atténuation d'horizons

Reconsidérer la notion d'horizons

### Idée

Définir l'horizon en fonction de l'intérêt des coups joués et non en fonction du nombre de coups.

### Principes

On part par exemple avec une profondeur  $SX = 10.p$ , où  $p$  est la profondeur au sens habituel.

- ▶ Un coup « moyen » diminue  $SX$  de 10.
- ▶ Un coup intéressant (prise de pièce, échec) ne diminue  $SX$  que de 2 ou 3.
- ▶ Un coup peu intéressant diminue beaucoup  $SX$  (de l'ordre de 35).

**Résultat** : les coups intéressants seront explorés plus profondément.

## Analyse des parties

Passer du temps là où il faut

Concentrer l'effort de recherche dans les *zones critiques*

- ▶ Si tous les débuts de partie se ressemblent, ne pas y consacrer trop de temps (ex : jouer au hasard pour le jeu des *amazones*)

## Analyse des parties

Passer du temps là où il faut

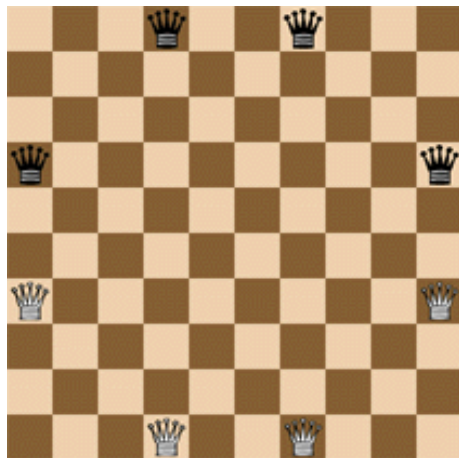
Concentrer l'effort de recherche dans les *zones critiques*

- ▶ Si tous les débuts de partie se ressemblent, ne pas y consacrer trop de temps (ex : jouer au hasard pour le jeu des *amazones*)
- ▶ Passer plus de temps au milieu de partie (ex Awalé) car les fins de partie ne demandent pas beaucoup de temps (élagage très efficace dans l'Awalé).

Mais seule une connaissance profonde du jeu permet d'affiner les moments où la recherche de bon coups est la plus critique.

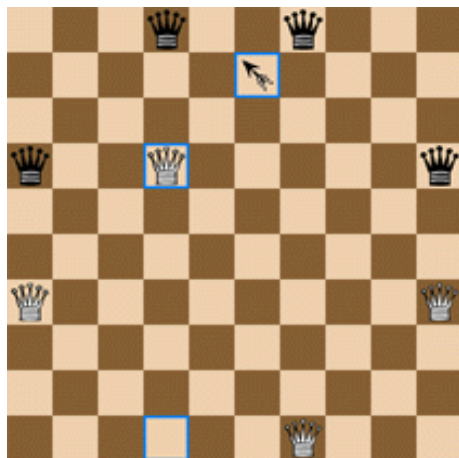
## Exemple sur les Amazones

Comment gérer l'ouverture ?



## Exemple sur les Amazones

Comment gérer l'ouverture ?



## Analyse des parties

Passer du temps là où il faut

### 3 phases communes à tous les jeux

- ▶ Début de partie
- ▶ Milieu de partie
- ▶ Fin de partie

Un bon programme de jeu doit avoir un comportement fondamentalement différent dans chacune de ces trois phases.

## Analyse des parties

Passer du temps là où il faut

### 3 phases communes à tous les jeux

- ▶ Début de partie
- ▶ Milieu de partie
- ▶ Fin de partie

Un bon programme de jeu doit avoir un comportement fondamentalement différent dans chacune de ces trois phases.

### Encore une heuristique

Il faut donc avoir une heuristique de haut niveau permettant de choisir entre les différentes heuristiques liées aux phases du jeu.

## Agir sur les heuristiques

On peut faire évoluer la valeur de la fonction heuristique suivant l'avancement dans le jeu.

### Exemples

- ▶ Awalé : faire des greniers en début de partie, puis simplement compter les graines en fin de partie
- ▶ Othello : Prendre des cases stratégiques, puis compter les pions.
- ▶ ...

La transition entre les différentes fonctions heuristiques doit être douce pour que l'horizon évolue doucement.



## Début de partie : bibliothèques d'ouverture

Pour concevoir une bibliothèque, deux solutions :

- ▶ Utilisation d'un expert (recopier des ouvertures dans la littérature).

## Début de partie : bibliothèques d'ouverture

Pour concevoir une bibliothèque, deux solutions :

- ▶ Utilisation d'un expert (recopier des ouvertures dans la littérature).
- ▶ Construction de ses propres bibliothèques. Utile lorsque le jeu n'a pas déjà été grandement étudié.

## Début de partie : bibliothèques d'ouverture

Pour concevoir une bibliothèque, deux solutions :

- ▶ Utilisation d'un expert (recopier des ouvertures dans la littérature).
- ▶ Construction de ses propres bibliothèques. Utile lorsque le jeu n'a pas déjà été grandement étudié.
  - ▶ Lancement d' $\alpha\beta$  à grande profondeur (extension hors-ligne de l'horizon des premiers coups)

## Début de partie : bibliothèques d'ouverture

Pour concevoir une bibliothèque, deux solutions :

- ▶ Utilisation d'un expert (recopier des ouvertures dans la littérature).
- ▶ Construction de ses propres bibliothèques. Utile lorsque le jeu n'a pas déjà été grandement étudié.
  - ▶ Lancement d' $\alpha\beta$  à grande profondeur (extension hors-ligne de l'horizon des premiers coups)
  - ▶ Faire jouer différentes ouvertures contre le programme lui-même

## Début de partie : bibliothèques d'ouverture

Pour concevoir une bibliothèque, deux solutions :

- ▶ Utilisation d'un expert (recopier des ouvertures dans la littérature).
- ▶ Construction de ses propres bibliothèques. Utile lorsque le jeu n'a pas déjà été grandement étudié.
  - ▶ Lancement d' $\alpha\beta$  à grande profondeur (extension hors-ligne de l'horizon des premiers coups)
  - ▶ Faire jouer différentes ouvertures contre le programme lui-même

### Comment stocker la bibliothèque ?

- ▶ Soit sous forme d'automate
- ▶ Soit sous forme de tables de transposition

## Début de partie : bibliothèques d'ouverture

Pour concevoir une bibliothèque, deux solutions :

- ▶ Utilisation d'un expert (recopier des ouvertures dans la littérature).
- ▶ Construction de ses propres bibliothèques. Utile lorsque le jeu n'a pas déjà été grandement étudié.
  - ▶ Lancement d' $\alpha\beta$  à grande profondeur (extension hors-ligne de l'horizon des premiers coups)
  - ▶ Faire jouer différentes ouvertures contre le programme lui-même

### Comment stocker la bibliothèque ?

- ▶ Soit sous forme d'automate
- ▶ Soit sous forme de tables de transposition

Ne pas oublier d'introduire du hasard...

## Milieu de partie

Domaine de  $\alpha\beta$

## Milieu de partie

Domaine de  $\alpha\beta$  avec tous les mécanismes vus :

- ▶ Recherche de coups profond
- ▶ Iterative Deepening
- ▶ Horizon de coups intéressants
- ▶ ...



## Fin de partie

Le domaine de la puissance brute

Généralement, le jeu est très contraint.

## Fin de partie

### Le domaine de la puissance brute

Généralement, le jeu est très contraint. Le facteur de branchement de l'arbre de jeu est réduit.

## Fin de partie

### Le domaine de la puissance brute

Généralement, le jeu est très contraint. Le facteur de branchement de l'arbre de jeu est réduit. On essaye d'aller jusqu'au bout de l'arbre de recherche pour trouver une stratégie gagnante.

## Fin de partie

### Le domaine de la puissance brute

Généralement, le jeu est très contraint. Le facteur de branchement de l'arbre de jeu est réduit. On essaye d'aller jusqu'au bout de l'arbre de recherche pour trouver une stratégie gagnante.

- ▶ Les heuristiques sont généralement réduites à « gagné » « perdu ».
- ▶ Il faut penser à augmenter considérablement l'horizon.