

Jeux d'accessibilité à deux joueurs

Fabrice Lembrez - PSI*

Lycée Pierre de Fermat

Plan

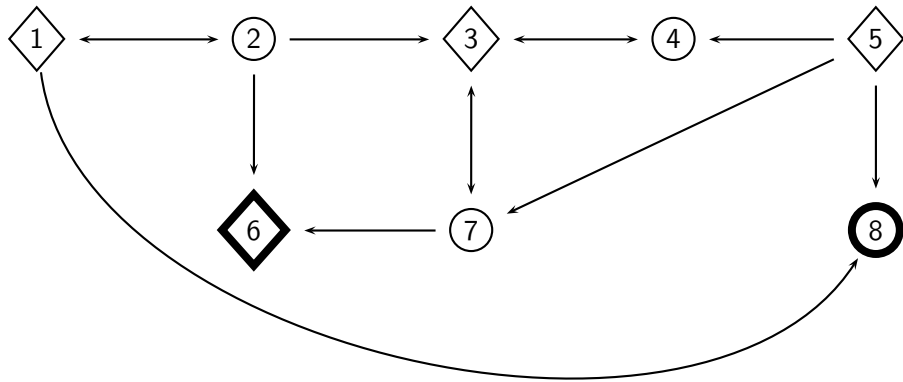
- 1 Description
 - Graphe associé
 - Stratégies
- 2 Recherche de stratégies gagnantes
- 3 Analyse de jeux complexes

Description par un graphe

On représente le jeu et les choix de stratégie à l'aide d'un graphe des "états du jeu". Voici les hypothèses "usuelles" (attention un sujet de concours pourrait introduire des variantes !)

- il y a deux joueurs 1 et 2
- il y a un nombre fini d'états
- les états finaux (sans flèche sortante) : victoire pour 1, pour 2, éventuellement "nul"
- les autres états sont "contrôlés" par J1 ou par J2
c'est le joueur qui a la main et choisit le coup suivant
- les états pour J1 ou J2 forment un graphe "biparti" : arêtes de 1 vers 2 ou 2 vers 1
interprétation : la main passe d'un joueur à l'autre
- match nul si partie infinie ou si on atteint l'état éventuel "nul"
- l'information est parfaite : les deux joueurs connaissent le contenu des cases

Exemple



Sur le schéma, on définit alors la notion de partie, de partie gagnante pour J1 ou J2.

Autres exemples plus sophistiqués : morpion (avec des états "match nul"), échecs...

Plan

- 1 Description
 - Graphe associé
 - Stratégies
- 2 Recherche de stratégies gagnantes
- 3 Analyse de jeux complexes

Stratégies

Ici on ne considère que des stratégies "sans mémoire", c'est-à-dire que si le jeu retrouve le même état, on joue de la même façon.

Déf - Stratégie (sans mémoire)

Pour le joueur J1, se donner une stratégie c'est faire un choix de réponse pour chaque état de jeu contrôlé par J1. Mathématiquement, c'est une fonction définie des états de jeu contrôlés par J1 vers ceux contrôlés par J2.

Un joueur peut fixer une stratégie et la suivre : quoi que joue l'autre il y a une réponse prévue à l'avance.

Stratégie gagnante, position gagnante

Déf - Stratégie gagnante, position gagnante

Soit P_0 un état du jeu. On dit qu'il y a une stratégie gagnante pour le joueur J1 à partir de P_0 s'il existe une stratégie telle que toutes les parties issues de P_0 et respectant la stratégie conduisent à la victoire de J1. Dans ce cas on dit également que P_0 est une position gagnante pour J1.

Cela signifie : à partir d'une telle position, si J1 joue intelligemment il gagne forcément.

Stratégie gagnante, position gagnante

Déf - Stratégie gagnante, position gagnante

Soit P_0 un état du jeu. On dit qu'il y a une stratégie gagnante pour le joueur J1 à partir de P_0 s'il existe une stratégie telle que toutes les parties issues de P_0 et respectant la stratégie conduisent à la victoire de J1. Dans ce cas on dit également que P_0 est une position gagnante pour J1.

Cela signifie : à partir d'une telle position, si J1 joue intelligemment il gagne forcément.

Une position initiale peut être

- gagnante pour J1,
- ou alors gagnante pour J2,
- ou alors pour aucun des deux.

Dans ce dernier cas si les deux joueurs jouent bien, la partie sera nulle (éventuellement infinie).

Plan

- 1 Description
- 2 Recherche de stratégies gagnantes
 - Calcul d'attracteurs
 - Algorithme de calcul
 - Variantes et remarques
- 3 Analyse de jeux complexes

Attracteurs

Si on connaît déjà un certain nombre de positions gagnantes pour J_1 , formant un ensemble Z , on peut en fabriquer de nouvelles :

- un état contrôlé par J_2 qui ne débouche que sur des éléments de Z est gagnant pour J_1
- un état contrôlé par J_1 qui débouche sur au moins un élément de Z est gagnant pour J_1

On peut créer ainsi des positions gagnantes de proche en proche. On va construire formellement les ensembles successifs obtenus par ce procédé.

Déf - Attracteur pour le joueur 1

Soit $G = (S, A)$ un graphe de jeu (une "arène"), biparti (sommets S_1 / S_2 à tour de rôle). On appelle Att_0 l'ensemble des états finaux pour J1. On construit successivement les ensembles Att_k en posant

$$Att_{j+1} = Att_j \cup V_{j+1} \text{ avec}$$

$$V_{2k+1} = \{x \in S_2, \text{ non final}, \forall x' \in S_1, (x, x') \in A \Rightarrow x' \in Att_{2k}\}$$

$$V_{2k+2} = \{x \in S_1, \exists x' \in S_2, (x, x') \in A, x' \in Att_{2k+1}\}$$

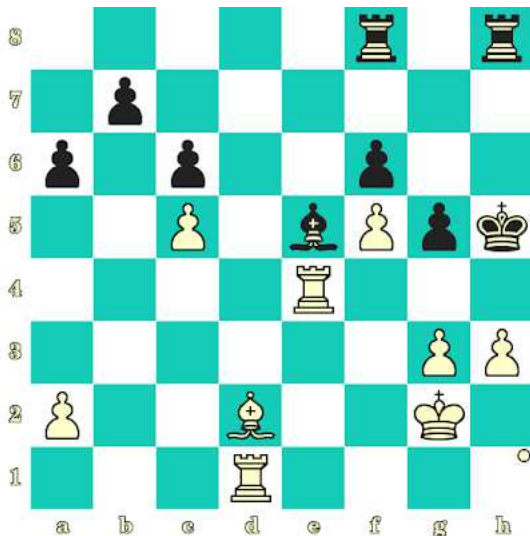
Et finalement la réunion croissante de tous ces ensembles, qui est appelée attracteur Att de J1. On peut parler du rang d'un élément $x \in Att$: c'est le premier k tel que $x \in Att_k$

Bien sûr la réunion croissante se stabilise en K étapes (graphe fini).

Attracteur et rang

Les Blancs jouent et
matent en 2 coups.

Anatoly Karpov vs
Piotr Mickiewicz,
Koszalin, 1997



Attracteurs et positions gagnantes

Lien entre attracteur et positions gagnantes

L'attracteur de J1 est l'ensemble des positions gagnantes pour J1.

Attracteurs et positions gagnantes

Lien entre attracteur et positions gagnantes

L'attracteur de J_1 est l'ensemble des positions gagnantes pour J_1 .

Le premier sens (tout point de l'attracteur est gagnant) suit la construction par récurrence les éléments de $\mathcal{A}_k$ sont gagnants de rang $\leq k$.

En effet un élément de V_{2k+1} est gagnant car tout successeur est gagnant de rang $\leq k$. Ainsi en choisissant un bon successeur, un élément de V_{2k+2} est gagnant de rang $\leq k + 1$.

Attracteurs et positions gagnantes

Lien entre attracteur et positions gagnantes

L'attracteur de J1 est l'ensemble des positions gagnantes pour J1.

Pour la réciproque, on prouve plutôt qu'un point hors de l'attracteur ne peut pas être gagnant. On s'appuie sur des remarques qui portent cette fois sur l'attracteur vu comme un tout

- 1 si x est contrôlé par J1 et pas dans l'attracteur, aucun de ses successeurs n'est dans l'attracteur.
- 2 si x est contrôlé par J2 et pas dans l'attracteur, il possède au moins un successeur qui n'est pas dans l'attracteur.

Avec ces éléments cette fois c'est J2 qui dispose d'une stratégie pour ne jamais rentrer dans l'attracteur. Et donc ne jamais arriver à une position de victoire pour J1.

Illustration concrète : jeu de chomp

Soit une tablette de chocolat (parfum pomme verte) dont le carré supérieur gauche est empoisonné. Adam et Eve jouent à tour de rôle. Ils choisissent un carré de chocolat et le mangent, ainsi que tous ceux qui sont à droite ou au-dessous.

Qui va mourir empoisonné ?

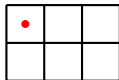
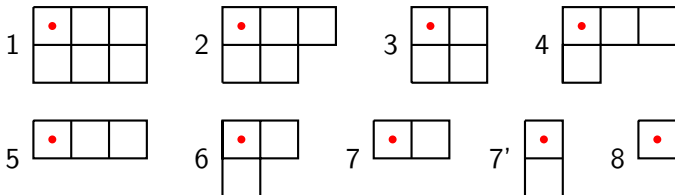


Illustration concrète : jeu de chomp

Soit une tablette de chocolat (parfum pomme verte) dont le carré supérieur gauche est empoisonné. Adam et Eve jouent à tour de rôle. Ils choisissent un carré de chocolat et le mangent, ainsi que tous ceux qui sont à droite ou au-dessous.

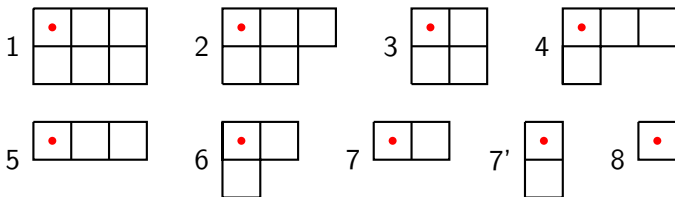
Qui va mourir empoisonné ?



On a dessiné 7 et 7' pour visualiser l'évolution de la plaque mais c'est le même état.

Ce n'est pas naturellement un jeu biparti mais on peut s'y ramener en "dupliquant" certains états : "la plaque est comme ceci est c'est à J1 de jouer".

Illustration concrète : jeu de chomp



On trace le graphe de ce jeu, en considérant que les états doivent intégrer l'information "à qui de jouer ?".

Calculer l'attracteur du premier joueur (rond) et vérifier qu'il doit gagner.

Illustration concrète : jeu de chomp

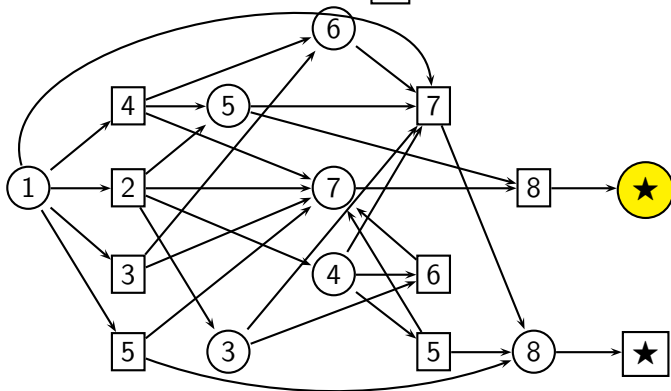
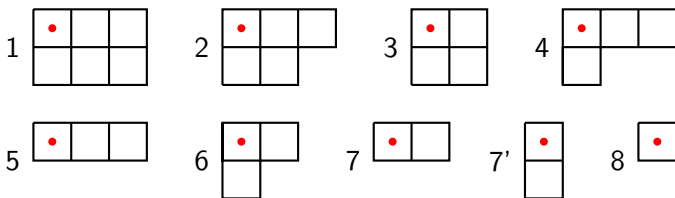


Illustration concrète : jeu de chomp

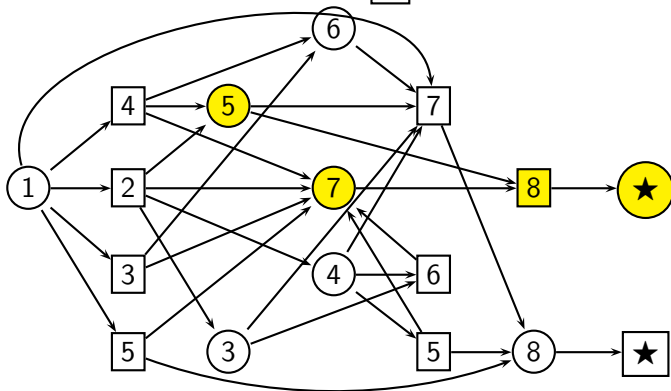
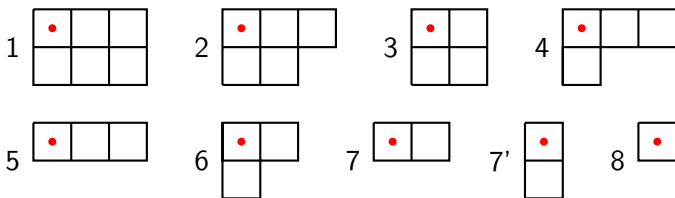


Illustration concrète : jeu de chomp

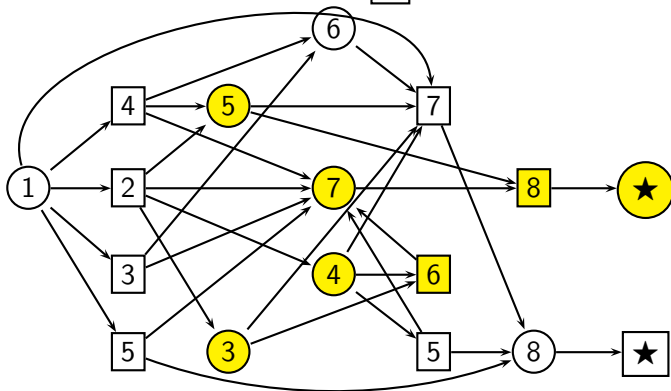
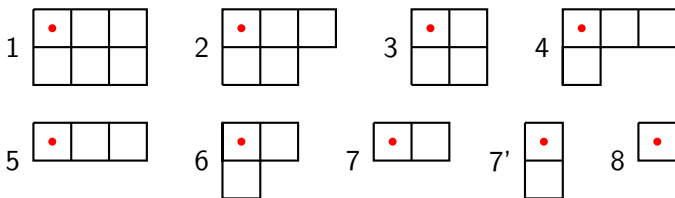


Illustration concrète : jeu de chomp

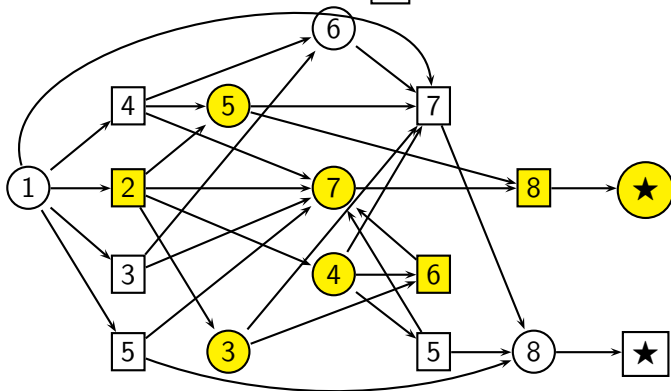
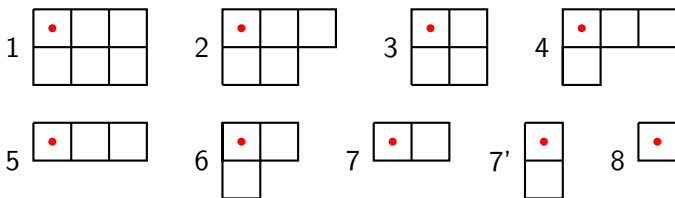
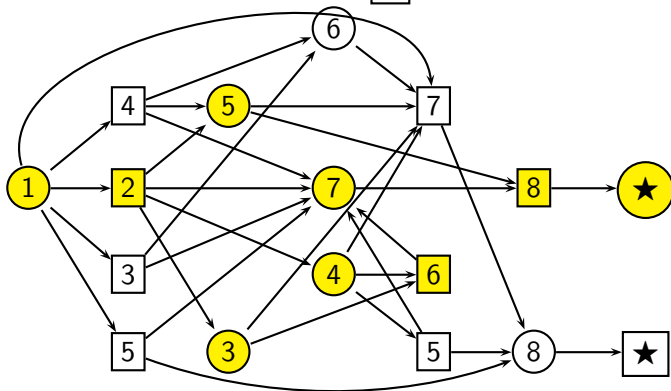
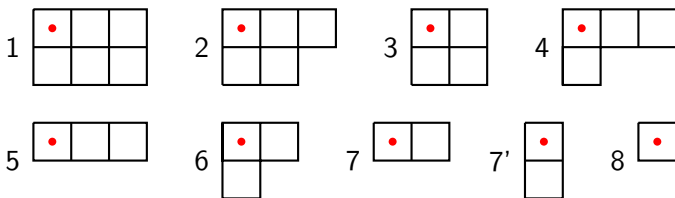


Illustration concrète : jeu de chomp



Plan

- 1 Description
- 2 Recherche de stratégies gagnantes
 - Calcul d'attracteurs
 - Algorithme de calcul
 - Variantes et remarques
- 3 Analyse de jeux complexes

Traitement algorithmique naïf

On calcule l'attracteur de J_1 en calquant la définition :

- trouver les états finaux appartenant à $J_1 = \text{rang } 0$.
- puis **chercher et stocker les nouveaux, rang par rang**
- pour les étapes impaires, simple recherche d'antécédents **non déjà trouvés**
- pour les étapes paires, recherche d'antécédents non déjà trouvés + condition "non final, tous les successeurs sont dans l'attracteur".

Conteneurs adaptés

Traitement algorithmique naïf

On calcule l'attracteur de $J1$ en calquant la définition :

- trouver les états finaux appartenant à $J1 = \text{rang } 0$.
- puis **chercher et stocker les nouveaux, rang par rang**
- pour les étapes impaires, simple recherche d'antécédents **non déjà trouvés**
- pour les étapes paires, recherche d'antécédents non déjà trouvés + condition "non final, tous les successeurs sont dans l'attracteur".

Conteneurs adaptés

- on peut construire une liste d'éléments à ajouter à l'étape k
- mais cette liste est écrasée à chaque étape,
- il faut aussi savoir qui est dans l'attracteur, à quel rang
- on peut même mémoriser quelle stratégie choisir

Voir TD pour la mise en œuvre. En tout cas, **il vaut mieux que la recherche d'antécédents soit facile.**

Un gros défaut

- On reteste plusieurs fois les successeurs d'un même sommet pour voir s'ils sont tous dans l'attracteur, au fur et à mesure que l'attracteur grandit.

Exemple : le "2" sur le jeu de Chomp.

Correctif : mémoriser le nombre de successeurs trouvés dans l'attracteur et le mettre à jour

Traitement algorithmique efficace et plus général

On ne sépare pas les étapes pair/impair : c'est un traitement sommet par sommet.

Choix de mémorisation

- pour chaque sommet, la liste des prédécesseurs
- pour chaque sommet, le nombre de successeurs $S[x]$ hors de l'attracteur

Intérêt : on saura quand ce nombre passe à 0

Mise en œuvre (cf TD)

- initialisations
- parcours en largeur ou en profondeur
- on déclenche une visite à un prédécesseur non visité : ceux contrôlés par J1, ceux contrôlés par J2 ayant $S[x]=0$
- penser à mettre à jour $S[x]$

Avantage de l'algorithme : complexité temporelle en $O(n + m)$ (nombre de sommets et nombre d'arêtes).

Plan

- 1 Description
- 2 Recherche de stratégies gagnantes
 - Calcul d'attracteurs
 - Algorithme de calcul
 - Variantes et remarques
- 3 Analyse de jeux complexes

Remarques

Quelques remarques pour éviter d'être pris au dépourvu lors des concours.

Le fait que le jeu soit "biparti" (alternance des deux joueurs) n'a rien d'essentiel. On a quasiment les mêmes résultats sans cela.

On peut présenter le concept de "victoire" autrement

- pour un "jeu d'accessibilité" au sens strict on ne parle que des positions de victoire de J1 et soit il réussit soit il échoue à les atteindre (et on dit que J2 est vainqueur).
- pour un "jeu d'évitement" au sens strict on cherche au contraire à éviter certaines positions.

Avec cette façon de parler

- victoire dans le "jeu d'accessibilité" = victoire de J1 dans le "jeu symétrique"
- échec dans le "jeu d'accessibilité" = victoire de J2 ou nul dans le "jeu symétrique"

Plan

- 1 Description
- 2 Recherche de stratégies gagnantes
- 3 Analyse de jeux complexes
 - Limites de l'exploration exhaustive
 - Introduction d'une heuristique
 - Minmax
 - Backtracking

Limites de l'exploration exhaustive

Le calcul d'attracteur a un aspect arborescent, mais il y a deux déficits d'information

- pas de conseil stratégique pour exploiter au mieux une position non gagnante (c'est le mode "tout ou rien").
- l'exploration complète est souvent impossible (elle revêt un caractère exponentiel)

Cela n'a d'intérêt que pour quelques jeux très simples, ou alors pour des jeux plus sophistiqués, au moment où l'on arrive aux "derniers coups".

Limites de l'exploration exhaustive

En fait sur un jeu tel que les échecs, Othello,... il y a

- une phase d'ouverture, où l'ordinateur va exploiter une base de données d'ouvertures classiques, avec des probabilités de succès fondées sur une analyse statistique de parties réelles
- une phase intermédiaire où l'on ne peut analyser que "quelques coups en avant"
- la phase finale où l'on peut refaire une analyse complète par force brute.

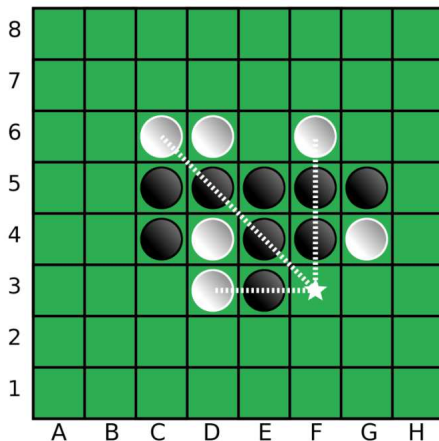
On va s'intéresser à cette phase intermédiaire...

Plan

- 1 Description
- 2 Recherche de stratégies gagnantes
- 3 Analyse de jeux complexes
 - Limites de l'exploration exhaustive
 - Introduction d'une heuristique
 - Minmax
 - Backtracking

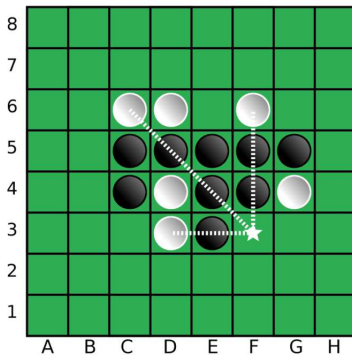
Fonction heuristique d'évaluation

A chaque position du jeu on attribue une valeur stratégique à partir d'une fonction d'évaluation ad hoc. Cela dépend complètement du jeu considéré. Prenons l'exemple d'Othello/Reversi



Exemple : Othello/Reversi

- 1- Evaluation naïve : nombre de pions blancs d'avance sur les noirs.
- 2- Pondérer les positions : "rester au centre", "atteindre les bords", "occuper un coin"
- 3- Raffiner : tenir compte de la configuration des cases voisines, et des cases laissées libres à l'adversaire.



Exemple : Othello/Reversi

- 1- Evaluation naïve : nombre de pions blancs d'avance sur les noirs.
- 2- Pondérer les positions : "rester au centre", "atteindre les bords", "occuper un coin"
- 3- Raffiner : tenir compte de la configuration des cases voisines, et des cases laissées libres à l'adversaire.

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

Fonction heuristique d'évaluation

Aux échecs, les pièces sont considérées comme de valeur différente (de 1 pour le pion à 9 pour la dame), et il y a énormément de remarques à prendre en compte sur leur position en termes de menace, liberté de mouvement, occupation du centre, etc...

Bref il s'agit d'un domaine en soi et on se contente de supposer qu'une telle fonction d'évaluation existe.

Fonction heuristique d'évaluation

Aux échecs, les pièces sont considérées comme de valeur différente (de 1 pour le pion à 9 pour la dame), et il y a énormément de remarques à prendre en compte sur leur position en termes de menace, liberté de mouvement, occupation du centre, etc...

Bref il s'agit d'un domaine en soi et on se contente de supposer qu'une telle fonction d'évaluation existe.

Le problème étudié

Si on "regarde quelques coups en avant" et si on applique la fonction d'évaluation aux différentes positions possibles, quel coup privilégier ?

Plan

- 1 Description
- 2 Recherche de stratégies gagnantes
- 3 Analyse de jeux complexes
 - Limites de l'exploration exhaustive
 - Introduction d'une heuristique
 - **Minmax**
 - Backtracking

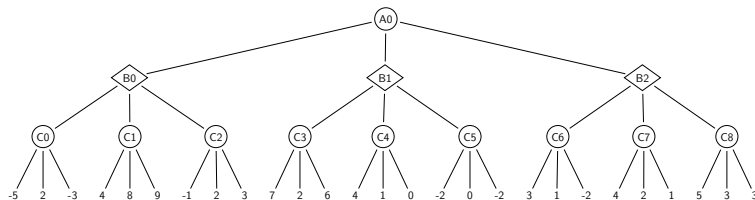
Minmax

Exemple d'arbre de décision pour un jeu à deux joueurs.

Les cases rondes sont contrôlées par le joueur 1, les carrées par le joueur 2.

Le joueur 1 a la main dans la situation A0 et il est capable d'anticiper les 3 prochains coups.

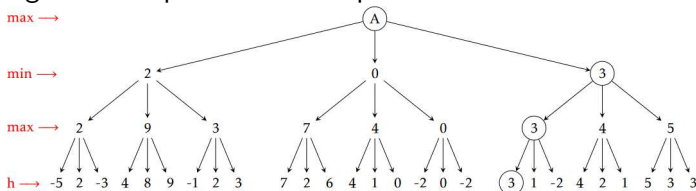
Sur la dernière ligne on a fait figurer la valeur des positions atteintes au bout de trois coups.



Minmax : Exemple

Exemple : Adam arrive à voir "3 coups en avant" (1er coup : c'est lui qui joue, deuxième coup Eve, et 3ème coup ce sera lui). Il sait aussi calculer l'heuristique sur les positions atteignables après ces 3 coups.

Il sait qu'au 3ème coup il cherchera à obtenir la meilleure valeur de l'heuristique parmi celles possibles. Puis qu'au deuxième coup Eve fera en sorte qu'il ait la moins bonne valeur parmi celles possibles compte tenu de son troisième coup. Et enfin au premier coup il cherchera la meilleure valeur atteignable compte tenu de ce qui va suivre.



Le minmax

Le joueur choisit le meilleur mouvement possible parmi ceux rendus possibles par l'adversaire.

Sachant que l'adversaire tirera de chaque coup les plus basses valeurs possibles, le joueur prend le max de ces min.

Concrètement on avance dans l'arbre de façon récursive en effectuant une alternance de maximisations/minimisations. La descente se produit jusqu'à une certaine hauteur ou jusqu'à atteindre une position finale.

Plan

- 1 Description
- 2 Recherche de stratégies gagnantes
- 3 Analyse de jeux complexes
 - Limites de l'exploration exhaustive
 - Introduction d'une heuristique
 - Minmax
 - Backtracking