

Jeux et intelligence artificielle

Eric Duchêne

Laboratoire LIRIS, IUT Lyon 1

4 janvier 2023



Jeux abstraits à deux joueurs : définition

Jeux abstraits à deux joueurs : définition

- 1 ou 2 joueurs exactement, qui jouent alternativement



Echecs



Tarot



Othello



Dames



Morpion



Petits chevaux



Go

Jeux abstraits à deux joueurs : définition

- 1 ou 2 joueurs exactement, qui jouent alternativement



Echecs



Tarot



Othello



Dames



Morpion



Petits chevaux



Go

Jeux abstraits à deux joueurs : définition

- 1 ou 2 joueurs exactement, qui jouent alternativement
- Information totale, pas de hasard



Echecs



Tarot



Othello



Dames



Morpion



Petits chevaux



Go

Jeux abstraits à deux joueurs : définition

- 1 ou 2 joueurs exactement, qui jouent alternativement
- Information totale, pas de hasard



Echecs



Tarot



Othello



Dames



Morpion



Petits chevaux



Go

Jeux abstraits à deux joueurs : définition

- 1 ou 2 joueurs exactement, qui jouent alternativement
- Information totale, pas de hasard
- Nombre fini de tours, toujours un gagnant ou nul



Echecs



Tarot



Othello



Dames



Morpion



Petits chevaux



Go

Jeux abstraits à deux joueurs : définition

- 1 ou 2 joueurs exactement, qui jouent alternativement
- Information totale, pas de hasard
- Nombre fini de tours, toujours un gagnant ou nul
- Vainqueur déterminé par le dernier coup, un score ou un motif réalisé



Echecs



Tarot



Othello



Dames



Morpion



Petits chevaux



Go

Un exemple : DOMINEERING

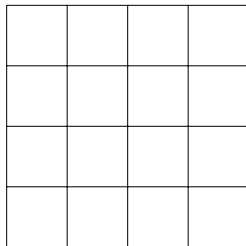
DOMINEERING

Une grille $n \times m$. A tour de rôle, Hector et Valérie posent un domino sur la grille.

Hector : horizontalement

Valérie : verticalement.

Le premier joueur bloqué perd.



Un exemple : DOMINEERING

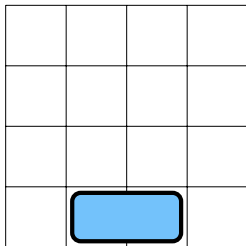
DOMINEERING

Une grille $n \times m$. A tour de rôle, Hector et Valérie posent un domino sur la grille.

Hector : horizontalement

Valérie : verticalement.

Le premier joueur bloqué perd.



Un exemple : DOMINEERING

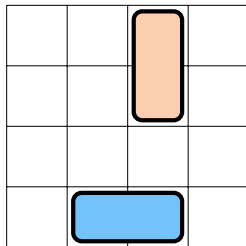
DOMINEERING

Une grille $n \times m$. A tour de rôle, Hector et Valérie posent un domino sur la grille.

Hector : horizontalement

Valérie : verticalement.

Le premier joueur bloqué perd.



Un exemple : DOMINEERING

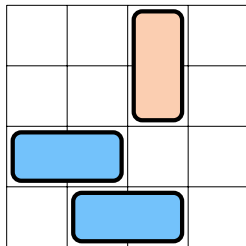
DOMINEERING

Une grille $n \times m$. A tour de rôle, Hector et Valérie posent un domino sur la grille.

Hector : horizontalement

Valérie : verticalement.

Le premier joueur bloqué perd.



Un exemple : DOMINEERING

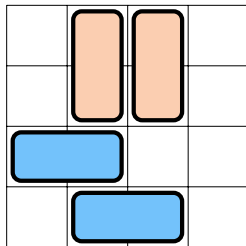
DOMINEERING

Une grille $n \times m$. A tour de rôle, Hector et Valérie posent un domino sur la grille.

Hector : horizontalement

Valérie : verticalement.

Le premier joueur bloqué perd.



Un exemple : DOMINEERING

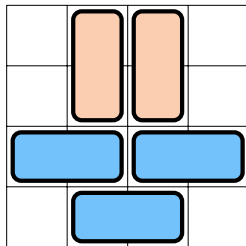
DOMINEERING

Une grille $n \times m$. A tour de rôle, Hector et Valérie posent un domino sur la grille.

Hector : horizontalement

Valérie : verticalement.

Le premier joueur bloqué perd.



Un exemple : DOMINEERING

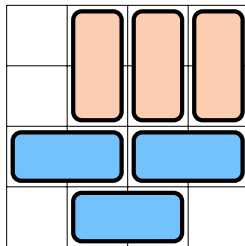
DOMINEERING

Une grille $n \times m$. A tour de rôle, Hector et Valérie posent un domino sur la grille.

Hector : horizontalement

Valérie : verticalement.

Le premier joueur bloqué perd.



Valérie gagne!

Problématiques du domaine

Etant donné un jeu, plusieurs problèmes associés.

Problématiques du domaine

Etant donné un jeu, plusieurs problèmes associés.

Problème 1 : Issue du jeu. Qui gagne?

Décider qui est le gagnant du jeu, en supposant les deux joueurs intelligents.

Problématiques du domaine

Etant donné un jeu, plusieurs problèmes associés.

Problème 1 : Issue du jeu. Qui gagne?

Décider qui est le gagnant du jeu, en supposant les deux joueurs intelligents.

Possibilités : Hector gagne, Valérie gagne, dépend de qui commence?

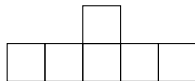
Problématiques du domaine

Etant donné un jeu, plusieurs problèmes associés.

Problème 1 : Issue du jeu. Qui gagne?

Décider qui est le gagnant du jeu, en supposant les deux joueurs intelligents.

Possibilités : Hector gagne, Valérie gagne, dépend de qui commence?



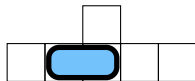
Problématiques du domaine

Etant donné un jeu, plusieurs problèmes associés.

Problème 1 : Issue du jeu. Qui gagne?

Décider qui est le gagnant du jeu, en supposant les deux joueurs intelligents.

Possibilités : Hector gagne, Valérie gagne, dépend de qui commence?



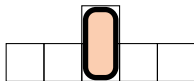
Problématiques du domaine

Etant donné un jeu, plusieurs problèmes associés.

Problème 1 : Issue du jeu. Qui gagne?

Décider qui est le gagnant du jeu, en supposant les deux joueurs intelligents.

Possibilités : Hector gagne, Valérie gagne, dépend de qui commence?



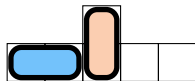
Problématiques du domaine

Etant donné un jeu, plusieurs problèmes associés.

Problème 1 : Issue du jeu. Qui gagne?

Décider qui est le gagnant du jeu, en supposant les deux joueurs intelligents.

Possibilités : Hector gagne, Valérie gagne, dépend de qui commence?



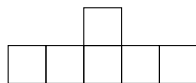
Problématiques du domaine

Etant donné un jeu, plusieurs problèmes associés.

Problème 1 : Issue du jeu. Qui gagne?

Décider qui est le gagnant du jeu, en supposant les deux joueurs intelligents.

Possibilités : Hector gagne, Valérie gagne, dépend de qui commence ?



H

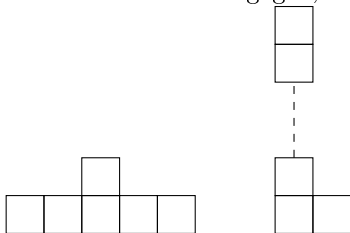
Problématiques du domaine

Etant donné un jeu, plusieurs problèmes associés.

Problème 1 : Issue du jeu. Qui gagne?

Décider qui est le gagnant du jeu, en supposant les deux joueurs intelligents.

Possibilités : Hector gagne, Valérie gagne, dépend de qui commence ?



H

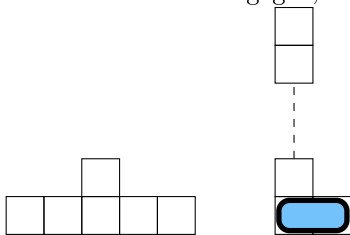
Problématiques du domaine

Etant donné un jeu, plusieurs problèmes associés.

Problème 1 : Issue du jeu. Qui gagne?

Décider qui est le gagnant du jeu, en supposant les deux joueurs intelligents.

Possibilités : Hector gagne, Valérie gagne, dépend de qui commence ?



H

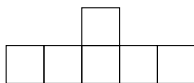
Problématiques du domaine

Etant donné un jeu, plusieurs problèmes associés.

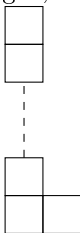
Problème 1 : Issue du jeu. Qui gagne?

Décider qui est le gagnant du jeu, en supposant les deux joueurs intelligents.

Possibilités : Hector gagne, Valérie gagne, dépend de qui commence ?



H



V

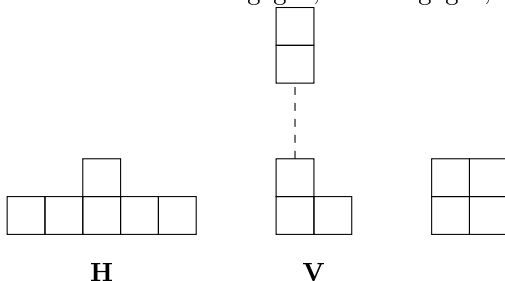
Problématiques du domaine

Etant donné un jeu, plusieurs problèmes associés.

Problème 1 : Issue du jeu. Qui gagne?

Décider qui est le gagnant du jeu, en supposant les deux joueurs intelligents.

Possibilités : Hector gagne, Valérie gagne, dépend de qui commence ?



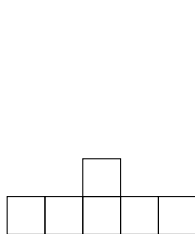
Problématiques du domaine

Etant donné un jeu, plusieurs problèmes associés.

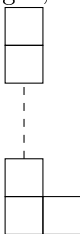
Problème 1 : Issue du jeu. Qui gagne?

Décider qui est le gagnant du jeu, en supposant les deux joueurs intelligents.

Possibilités : Hector gagne, Valérie gagne, dépend de qui commence ?



H



V



J1

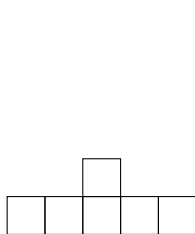
Problématiques du domaine

Etant donné un jeu, plusieurs problèmes associés.

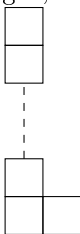
Problème 1 : Issue du jeu. Qui gagne?

Décider qui est le gagnant du jeu, en supposant les deux joueurs intelligents.

Possibilités : Hector gagne, Valérie gagne, dépend de qui commence ?



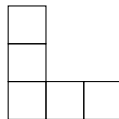
H



V



J1



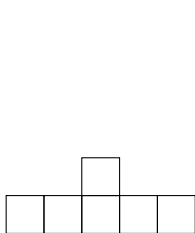
Problématiques du domaine

Etant donné un jeu, plusieurs problèmes associés.

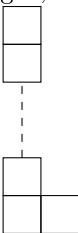
Problème 1 : Issue du jeu. Qui gagne?

Décider qui est le gagnant du jeu, en supposant les deux joueurs intelligents.

Possibilités : Hector gagne, Valérie gagne, dépend de qui commence ?



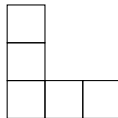
H



V



J1



J2

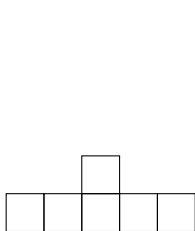
Problématiques du domaine

Etant donné un jeu, plusieurs problèmes associés.

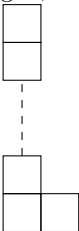
Problème 1 : Issue du jeu. Qui gagne?

Décider si un jeu est \mathcal{H} , \mathcal{V} , $\mathcal{J}1$ ou $\mathcal{J}2$.

Possibilités : Hector gagne, Valérie gagne, dépend de qui commence?



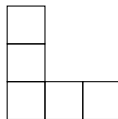
H



V



J1



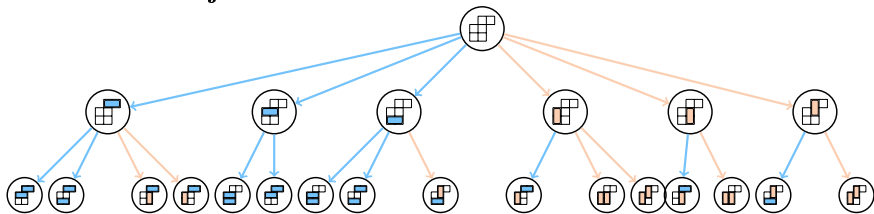
J2

Problème 1 : résolution

Outil : **arbre de jeu**

Problème 1 : résolution

Outil : arbre de jeu



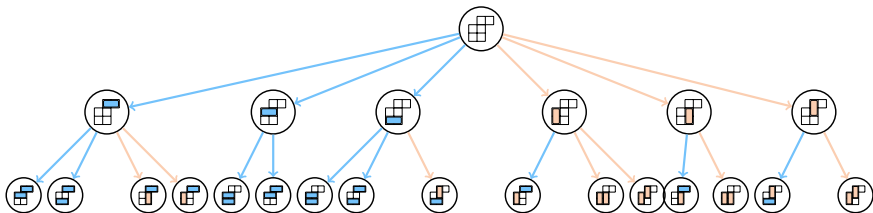
Problème 1 : résolution

Theorem (Berlekamp, Conway, Guy, 1981)

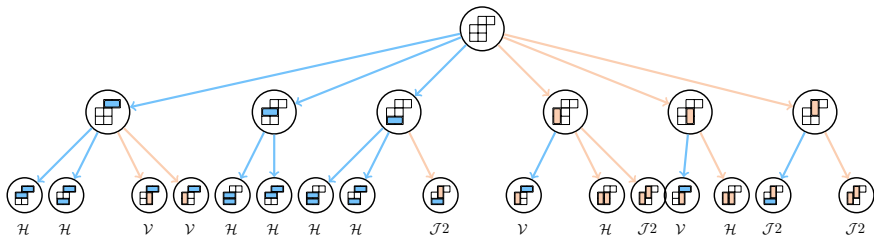
Pour un jeu dont le vainqueur est déterminé par le dernier coup, un noeud d'un arbre de jeu sera étiqueté :

- *$\mathcal{J}2$ si ses fils gauches sont tous $\mathcal{J}1$ ou \mathcal{V} et ses fils droits sont tous $\mathcal{J}1$ ou \mathcal{H} .*
- *$\mathcal{J}1$ si au moins un fils gauche est \mathcal{H} ou $\mathcal{J}2$ et au moins un fils droit est \mathcal{V} ou $\mathcal{J}2$.*
- *\mathcal{H} si au moins un fils gauche est \mathcal{H} ou $\mathcal{J}2$ et ses fils droits sont tous $\mathcal{J}1$ ou \mathcal{H} .*
- *\mathcal{V} si ses fils gauches sont tous \mathcal{V} ou $\mathcal{J}1$ et au moins un fils droit est \mathcal{V} ou $\mathcal{J}2$.*

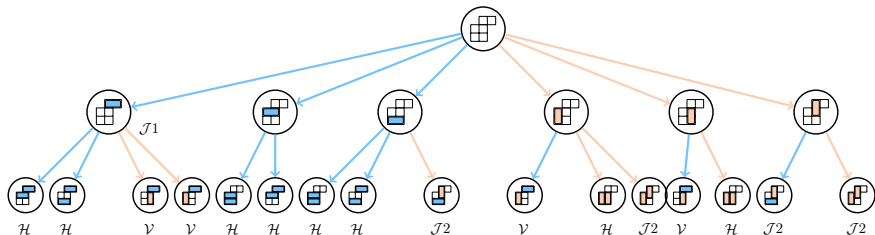
Problème 1 : résolution



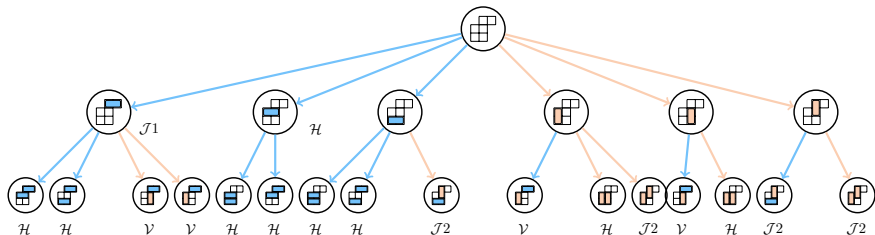
Problème 1 : résolution



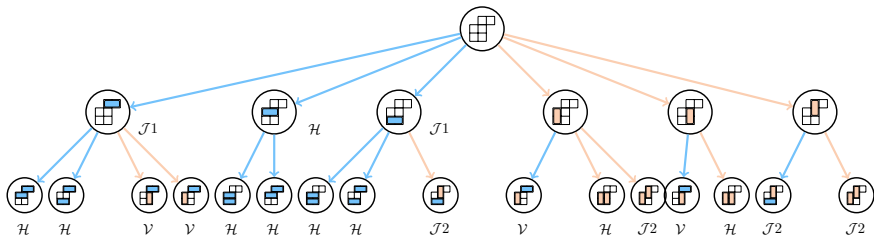
Problème 1 : résolution



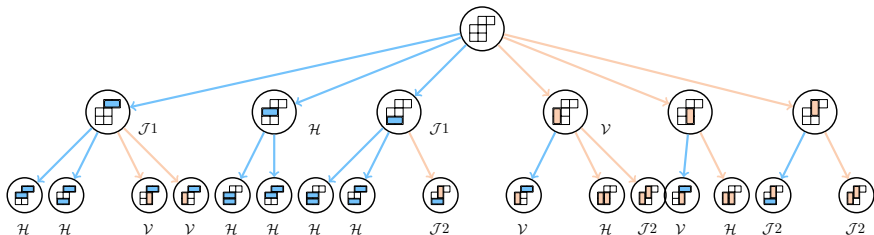
Problème 1 : résolution



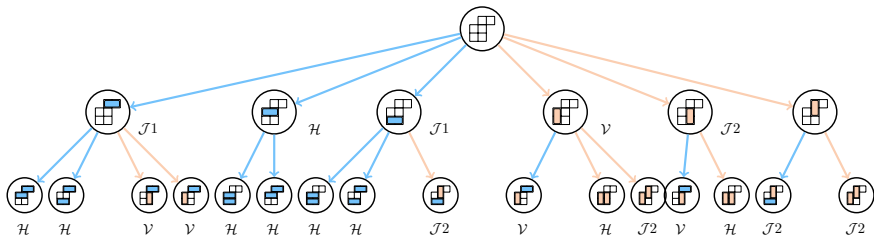
Problème 1 : résolution



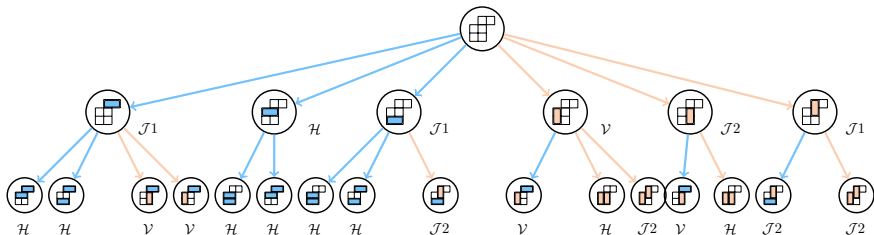
Problème 1 : résolution



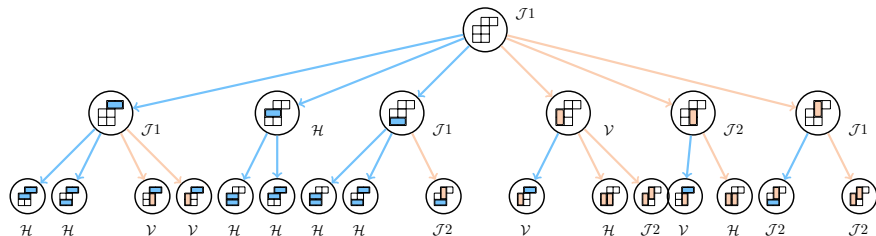
Problème 1 : résolution



Problème 1 : résolution



Problème 1 : résolution



On peut programmer ce calcul sur un ordinateur. Ça marche, mais la méthode est lourde et le temps de calcul peut être long

Problématiques du domaine

Problème 2 : Stratégie gagnante

Si un joueur a une stratégie gagnante, calculer son prochain coup gagnant quelle que soit les coups de son adversaire :

Problématiques du domaine

Problème 2 : Stratégie gagnante

Si un joueur a une stratégie gagnante, calculer son prochain coup gagnant quelle que soit les coups de son adversaire :

- calcul exact : atteint le score max du jeu à la fin.

Problématiques du domaine

Problème 2 : Stratégie gagnante

Si un joueur a une stratégie gagnante, calculer son prochain coup gagnant quelle que soit les coups de son adversaire :

- calcul exact : atteint le score max du jeu à la fin.
- calcul approché : maintien de l'issue par exemple (jeux à score, puzzles)

Problématiques du domaine

On s'intéresse à la complexité algorithmique des problèmes suivants :

Problème 1

Issue du jeu $(\mathcal{J}2, \mathcal{J}1, \mathcal{V}, \mathcal{H})$ ou valeur du meilleur score

Problème 2

Calcul du prochain coup gagnant

Problématiques du domaine

On s'intéresse à la complexité algorithmique des problèmes suivants :

Problème 1

Issue du jeu $(\mathcal{J}2, \mathcal{J}1, \mathcal{V}, \mathcal{H})$ ou valeur du meilleur score

Problème 2

Calcul du prochain coup gagnant

Il existe certains jeux où le problème 1 est facile, et le problème 2 très difficile.

Exemples de jeux dont l'issue est connue

CHOMP (Gale, 1974)

| | | | | | | | |
|--|--|--|--|--|--|--|--|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

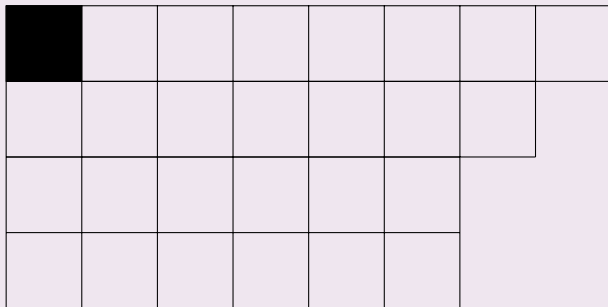
Exemples de jeux dont l'issue est connue

CHOMP (Gale, 1974)

| | | | | | | | |
|--|--|--|--|--|--|--|--|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

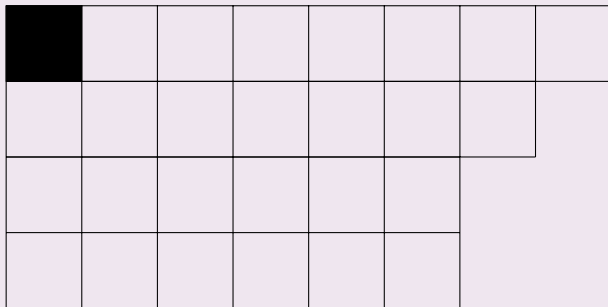
Exemples de jeux dont l'issue est connue

CHOMP (Gale, 1974)



Exemples de jeux dont l'issue est connue

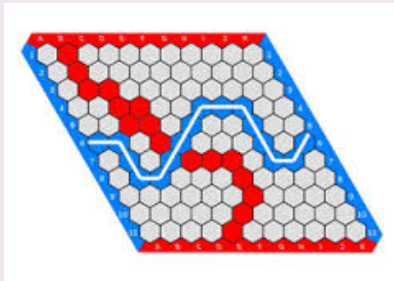
CHOMP (Gale, 1974)



CHOMP est $\mathcal{I}1$, mais problème 2 complexe.

Exemples de jeux dont l'issue est connue

HEX



HEX est $\mathcal{J}1$
Problème 2 difficile.

Critères de difficulté d'un jeu

Critères de difficulté d'un jeu (Fraenkel, 2006)

- Nombre de coups (facteur de branchement)
- Cycles
- Impartial/Partisan
- Décomposition en somme / Interactions
- Nombre de positions finales

Complexité connues

Quand calculer l'arbre de jeu en entier est possible...

Complexité connues

Quand calculer l'arbre de jeu en entier est possible...

DOMINEERING

Complexité inconnue en général.

Quand les tailles du plateau sont fixées :

| <i>n/m</i> | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
|------------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 2 | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H |
| 2 | V | 1 | 1 | H | V | 1 | 1 | H | V | 1 | 1 | H | 2 | 1 | 1 | H | H | 1 | 1 | H | H | H | 1 | H | H | H | 1 | H | H |
| 3 | V | 1 | 1 | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H |
| 4 | V | V | V | 1 | V | 1 | V | H | V | H | V | H | 2 | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H |
| 5 | V | H | V | H | 2 | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H |
| 6 | V | 1 | V | 1 | V | 1 | V | H | V | 1 | 1 | H | V | H | | H | | 1h | 1h | H | | H | 1h | H | 1h | H | 1h | H | 1h |
| 7 | V | 1 | V | H | V | H | 1 | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H |
| 8 | V | V | V | V | V | V | V | 1 | V | H | V | | V | | | 2h | | 1h | | H | | | | 1h | | | | | |
| 9 | V | H | V | H | V | H | V | H | 1 | H | 1h | H | 1h | H | 1h | H | 1h | H | 1h | H | 1h | H | H | H | H | H | H | H | H |
| 10 | V | 1 | V | V | V | 1 | V | V | V | 1 | | | V | | | | | | | 2h | | 1h | | | | 1h | | | |
| 11 | V | 1 | V | H | V | 1 | V | H | 1v | | 12 | H | | 1h | 1h | H | | 1h | 1h | H | | H | 1h | H | | H | 1h | H | |

Complexité connues



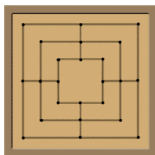
Puissance 4
Allen et Allis, 1988
 $\mathcal{I}1$ for 6×7 ,
ouvert si $n + m > 15$



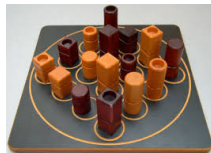
Dames
 8×8 : nul (Schaeffer, 2007)
 $n \times n$: PSPACE-difficile



Morpion
nul



Nine men's Morris
Gasser, 1993
nul



Quarto
Goossens, 1998
nul

Quand calculer l'issue est hors de portée...



Othello



Echecs



Go

Quand calculer l'issue est hors de portée...



Othello



Echecs



Go

Un nouvel objectif : résoudre un jeu de façon approchée

Concevoir un programme qui a les meilleures performances de jeu.

Pourquoi le Go et les échecs font-ils partie des jeux les plus difficiles ?

Echecs (Plateau 10×10)

Nombre de positions $\approx 10^{47}$

Facteur de branchement ≈ 35

Pourquoi le Go et les échecs font-ils partie des jeux les plus difficiles ?

Echecs (Plateau 10×10)

Nombre de positions $\approx 10^{47}$

Facteur de branchement ≈ 35

Go (plateau 19×19)

Nombre de positions $\approx 10^{171}$

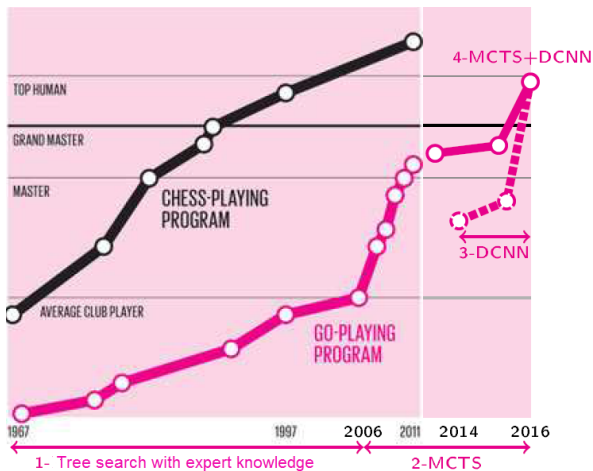
Facteur de branchement ≈ 250

Défi principal de l'IA pour les jeux

Comment gérer un arbre de jeu INCOMPLET ?

Choisir l'information partielle la plus pertinente.

Aperçu des performances de l'IA pour les échecs et le Go



De Schaeffer et al., 2014.

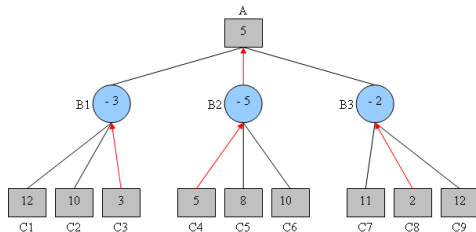
Arbre avec connaissance experte

Principe

Construction progressive de l'arbre de haut en bas

Construction en largeur (par niveaux horizontaux)

Fonction d'évaluation experte pour guider l'exploration



Algorithme MinMax

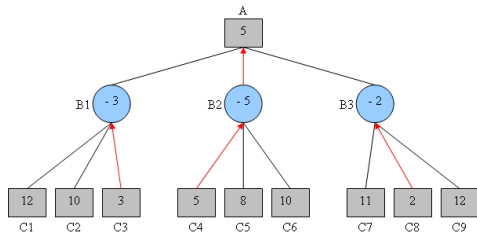
Arbre avec connaissance experte

Principe

Construction progressive de l'arbre de haut en bas

Construction en largeur (par niveaux horizontaux)

Fonction d'évaluation experte pour guider l'exploration



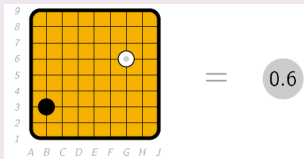
Algorithme MinMax

Améliorations bien connues : élagage $\alpha\beta$, tables de transposition...

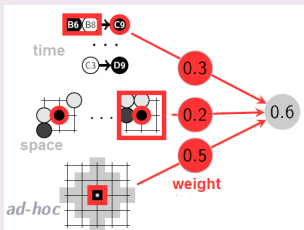
La fonction d'évaluation

Deux possibilités :

Evaluation directe depuis une position



Combinaison de paramètres



Succès et limites



- 1997 : Deep Blue (IBM) bat Kasparov aux échecs
- Deep Blue : 8000 paramètres, 10^8 états/sec, processeur dédié...
- Méthode peu réutilisable pour le go : difficile de trouver des bons paramètres, échelle

Jeux avec connaissance acquise

Absence de fonction d'évaluation performante :



MENACE : la première learning machine pour les jeux

Donald Michie : chercheur britannique en IA

MENACE (1961) : machine qui apprend à jouer toute seule au morpion

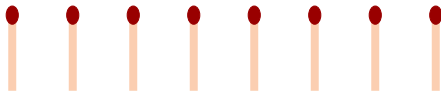
Machine avec des boîtes d'allumettes et des perles de couleur

Aucune connaissance au préalable et apprend à bien jouer au fur et à mesure



NIMBLE : une machine pour le jeu de NIM

Jeu de NIM : chaque joueur prend 1 ou 2 allumettes à chaque tour.
Vainqueur : celui qui prend la dernière.



- Chaque casier (avec une couleur) correspond à une position de jeu
- Chaque sac a au départ deux pions jaunes (-1) et rouges (-2)
- Un coup de la machine est un tirage au sort d'un pion dans le sac

NIMBLE : une machine pour le jeu de NIM

Principe de l'apprentissage par renforcement

- Simulation de parties aléatoires
- Des essais-erreur avec récompense ou punition qui changent les probabilités.
- Paramétrage important (valeur de renforcement, nombre de billes au départ...)

Ca se programme facilement :

<https://projet.liris.cnrs.fr/~mam/machine/>

NIMBLE : une machine pour le jeu de NIM

Principe de l'apprentissage par renforcement

- Simulation de parties aléatoires
- Des essais-erreur avec récompense ou punition qui changent les probabilités.
- Paramétrage important (valeur de renforcement, nombre de billes au départ...)

Ca se programme facilement :

<https://projet.liris.cnrs.fr/~mam/machine/>

Comment faire pour l'entraîner plus souvent ?

NIMBLE : une machine pour le jeu de NIM

Principe de l'apprentissage par renforcement

- Simulation de parties aléatoires
- Des essais-erreur avec récompense ou punition qui changent les probabilités.
- Paramétrage important (valeur de renforcement, nombre de billes au départ...)

Ca se programme facilement :

<https://projet.liris.cnrs.fr/~mam/machine/>

Comment faire pour l'entraîner plus souvent ? **Jouer contre elle-même**

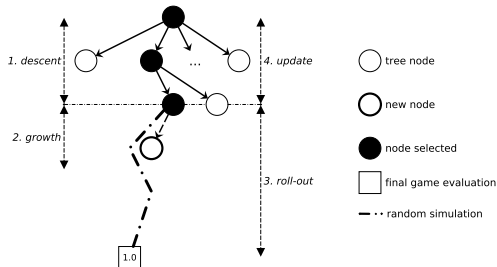
Monte Carlo Tree Search

Et sur des jeux plus compliqués?

Algorithme d'apprentissage le plus efficace pour les jeux : MCTS
(Bruegmann, 1993).

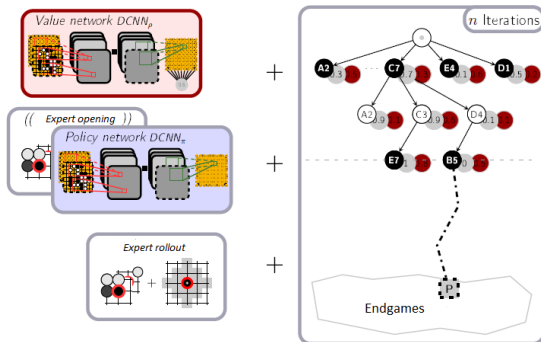
Principe de MCTS (Coulom, 2006)

- Apprentissage par renforcement
- Apprentissage profond (réseau de neurones)
- Construction incrémentale d'un arbre



Alpha Go (Google) : MCTS avec DCNN

DCNN : 2 réseaux de neurones profonds



De A. Fabbri, 2016.

Réseau politique : Apprentissage supervisé - intuition humaine

Réseau de valeurs : Auto-entraînement (apprentissage par renforcement) - évaluation statique

MCTS : exploitation - exploration de l'arbre.

Mars 2016 : Alpha Go VS Lee Sedol

Lee Sedol : 9 DAN PRO (meilleur humain)



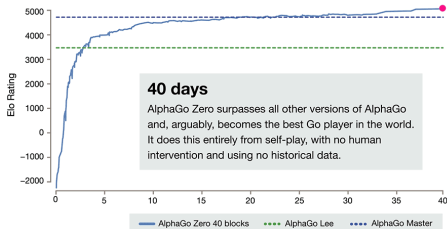
Prize money : 1000000 dollars

Mars 2016 : Alpha Go (V18) bat Lee Sedol 4-1

Octobre 2017 : Alpha Go Zero

Un pas énorme dans la réussite de l'IA. Les performances principales :

- A battu Alpha Go Lee 100 à 0.
- A appris à jouer à partir de rien (aucun apprentissage supervisé)
- En 3 jours, a le niveau du champion du monde de Go
- Petit nombre de processeurs (4 au lieu de 48)
- Performant pour d'autres jeux (échecs)



Autres applications de l'apprentissage par renforcement profond

Google DeepMind peut jouer comme un "superhumain" à 57 jeux Atari.



https://www.youtube.com/watch?v=V1eYniJ0Rnk&ab_channel=TwoMinutePapers

Si on relâche certaines contraintes des jeux

Information cachée, multijoueurs

Pluribus (2019) : réussit à battre plusieurs joueurs pro au Poker multijoueurs



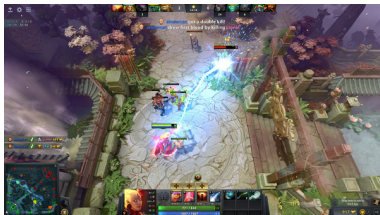
Si on relâche certaines contraintes des jeux

Information cachée, multijoueurs

Pluribus (2019) : réussit à battre plusieurs joueurs pro au Poker multijoueurs



OpenAI Five (2019) : bat l'équipe championne du monde de Dota 2 (entraînée pendant dix mois)



Autres domaines pour l'apprentissage par renforcement

Conduite autonome, réseaux d'ascenseurs, bras articulé, chatGPT...

