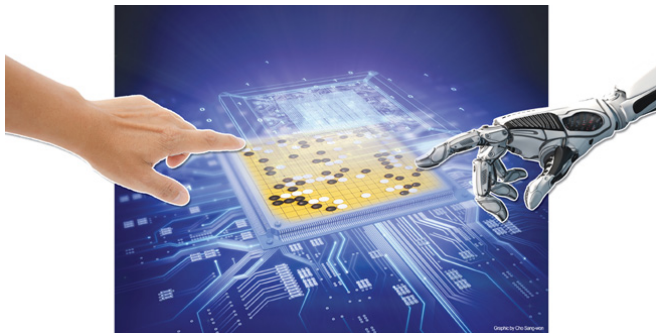




Jeu de Go et Exploration d'Arbre par Bandit

CentraleSupélec – Gif

IA et Jeu de Go



- 2016 : AlphaGo bat le meilleur joueur humain
- Combine des méthodes de **deep learning** avec une **exploration d'arbre par bandit**



Plan

- 1 L'IA et le Jeu de Go
 - Pourquoi une IA pour le Jeu de Go?
 - Le Jeu de Go
- 2 Avant l'Exploration d'Arbre par Bandit
- 3 Exploration d'Arbre par Bandit
- 4 Aller plus loin

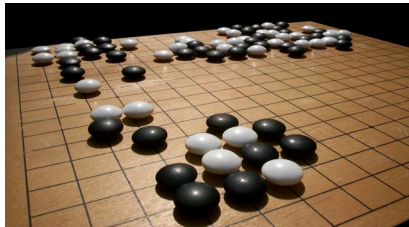
Pourquoi une IA pour un jeu ?



- Avoir une IA pour un jeu...
- Représentation des **problèmes de décision**
- **Environnement** bien défini : règles du jeu
- **Évaluation** facile : score
- **Challenge** de battre les humains



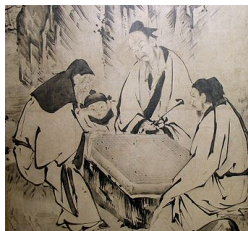
Pourquoi le jeu de Go ?



- un jeu de plateau qui a longtemps résisté aux IA
- règles **simples**
- méthodes classiques (alphabeta) inefficaces



Histoire



- aurait été inventé en chine en 2000 BC
- premiers écrits : 500 BC
- fait parti des 4 arts majeurs chinois :
peinture, calligraphie, guqin, go
- se répand en Asie dès 800 dans la noblesse
- aujourd'hui, environ 20 millions de joueurs



Matériel

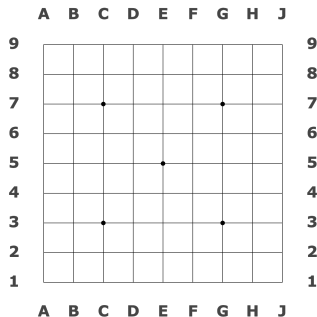
- plateau de jeu : Goban
- deux tailles 9x9 ou 19x19
- pierres noires et blanches





Règles : placement

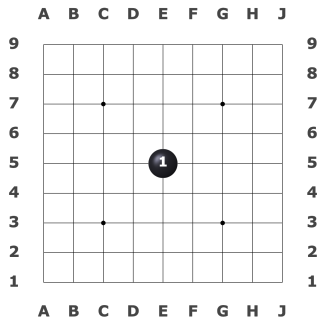
- Début de la partie : le plateau est vide
- Chaque joueur pose une pierre à tour de rôle
- Noir commence
- Pierres posées sur les intersections





Règles : placement

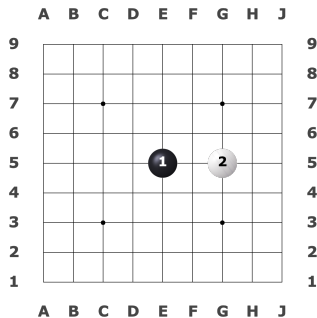
- Début de la partie : le plateau est vide
- Chaque joueur pose une pierre à tour de rôle
- Noir commence
- Pierres posées sur les intersections





Règles : placement

- Début de la partie : le plateau est vide
- Chaque joueur pose une pierre à tour de rôle
- Noir commence
- Pierres posées sur les intersections





Règles : chaînes et libertés

- Pierres reliées horizontalement ou verticalement : **une chaîne**
- Emplacements libres autour d'une chaîne : **liberté**



Règles : chaînes et libertés

- Pierres reliées horizontalement ou verticalement : **une chaîne**
- Emplacements libres autour d'une chaîne : **liberté**



Règles : chaînes et libertés

- Pierres reliées horizontalement ou verticalement : **une chaîne**
- Emplacements libres autour d'une chaîne : **liberté**



Règles : chaînes et libertés

- Pierres reliées horizontalement ou verticalement : **une chaîne**
- Emplacements libres autour d'une chaîne : **liberté**



Règles : chaînes et libertés

- Pierres reliées horizontalement ou verticalement : **une chaîne**
- Emplacements libres autour d'une chaîne : **liberté**



Règles : capture

- Pierres reliées horizontalement ou verticalement : **une chaine**
 - Emplacements libres autour d'une chaine : **liberté**
 - Enlever la dernière liberté d'une chaine : **capture**
- ➔ les pierres sont enlevées du plateau



Règles : fin de partie

les deux joueurs passent score



Règles : le ko

illustration histoire règle humain/ordinateur



Echelle de niveau



Plan

- 1 L'IA et le Jeu de Go
- 2 Avant l'Exploration d'Arbre par Bandit**
- 3 Exploration d'Arbre par Bandit
- 4 Aller plus loin



Présentation



Alpha beta



Découpage du plateau



Règles expertes



Échelle de niveau



Plan

- 1 L'IA et le Jeu de Go
- 2 Avant l'Exploration d'Arbre par Bandit
- 3 Exploration d'Arbre par Bandit**
 - Construction de l'Arbre
 - Problème de Bandit
 - Amélioration de l'Algorithme
- 4 Aller plus loin



Introduction du problème



Dans un casino, il y a plusieurs machines à sous différentes en terme de récompense.

- Comment répartir mes pièces entre les machines ?

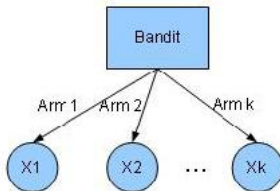
Autres problèmes similaires



- Essais cliniques : trouver le traitement qui fonctionne le mieux.
- Sélection d'un serveur dans un réseau : trouver le serveur avec le temps de réponse le plus faible.
- Publicité ciblée : trouver le type de pub qui intéressera le plus un utilisateur.
- ...

Ce sont des problèmes où on a plusieurs fois le même choix à effectuer. Le choix conduit à une récompense aléatoire.

Définition formelle



- un ensemble de bras $A = \{1, \dots, K\}$.
- chaque bras est associé à une distribution de probabilité X_k d'espérance μ_k .
- l'algorithme choisit un bras a à chaque pas de temps.
- le bandit retourne une récompense r : une réalisation de X_a .
- les tirages successifs sur un même bras sont indépendants et identiquement distribués.



Notations supplémentaires

- $T_i(n)$: le nombre de fois que le bras i a été sélectionné au pas de temps n .
- $\mu^* = \max_{1 \leq i \leq K} \mu_i$
- $\Delta_i = \mu^* - \mu_i$
- $\Delta = \min_{i: \Delta_i > 0} \Delta_i$

Objectif

Le but est d'optimiser le regret R_n défini comme suit :

$$R_n = \mu^* n - \mathbb{E} \sum_{j=1}^K T_j(n) \mu_j$$

$$R_n = \sum_{j=1}^K \Delta_j \mathbb{E}[T_j(n)]$$

Borne inférieure

Pour toute stratégie d'allocation et pour tout bras non optimal :

$$\mathbb{E}[T_j(n)] \geq \frac{\log n}{D(p_j||p^*)}$$

$$\text{où } D(p_j||p^*) = \int p_j \log \frac{p_j}{p^*}$$

On en déduit que le meilleur regret atteignable est en $\log(n)$.

[Lai and Robbins, 1985]



UCB

Principe de l'algorithme :

- A partir des informations disponibles au temps t , on calcule la borne de confiance supérieur (UCB) correspondant à chaque bras.
- On choisit le bras qui a la valeur UCB la plus grande.

[Auer and all, 2002]

UCB

Calcul de la valeur UCB pour le bras i au pas de temps t :

$$\hat{\mu}_{i,t-1} + \sqrt{\frac{3 \log(t)}{2T_i(t-1)}}$$

où $\hat{\mu}_{i,t-1}$ correspond à la moyenne empirique du bras i .

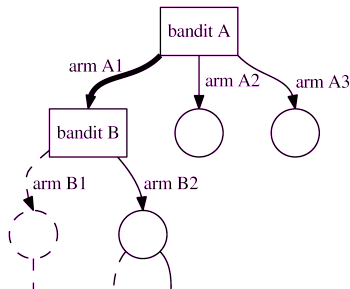


Borne sur le regret :

$$R_n \leq 6 * \sum_{i \neq i^*} \frac{\log(n)}{\Delta_i} + K\left(\frac{\pi^2}{3} + 1\right)$$

Descente dans l'arbre

La descente dans l'arbre se fait en considérant que chaque choix d'une branche est un problème de bandit.



UCB en pratique

- Ajout d'un paramètre p de contrôle de l'exploration :

$$\hat{\mu}_{i,t-1} + p \sqrt{\frac{\log(t)}{T_i(t-1)}}$$

- Ajout de connaissances a priori $C_i(t)$:

$$\hat{\mu}_{i,t-1} + p \sqrt{\frac{\log(t)}{T_i(t-1)}} + C_i(t)$$



AMAF



Ajout de Connaissances Expertes



Plan

- 1 L'IA et le Jeu de Go
- 2 Avant l'Exploration d'Arbre par Bandit
- 3 Exploration d'Arbre par Bandit
- 4 Aller plus loin**



Deep Learning

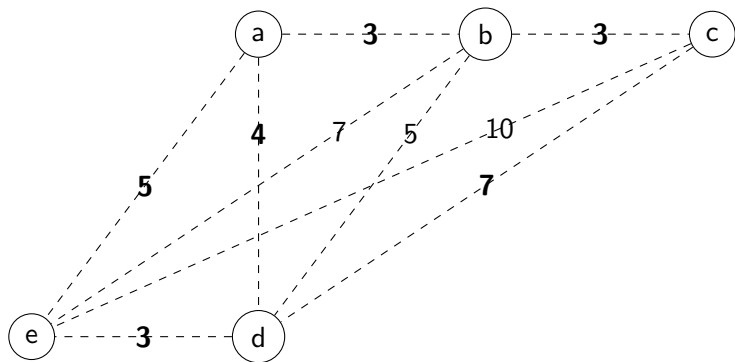


Autres applications



Conclusion

Exemple borne



Calcul de la borne

trajet déjà effectué = \emptyset

$$\underbrace{(3+4)}_a + \underbrace{(3+3)}_b + \underbrace{(3+7)}_c + \underbrace{(3+4)}_d + \underbrace{(3+5)}_e \bigg/ 2 = 19$$