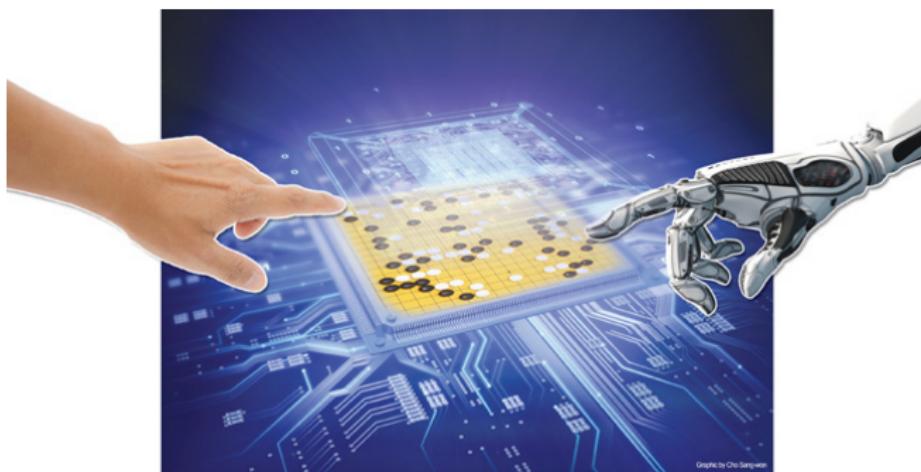




# Jeu de Go et Exploration d'Arbre par Bandit

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

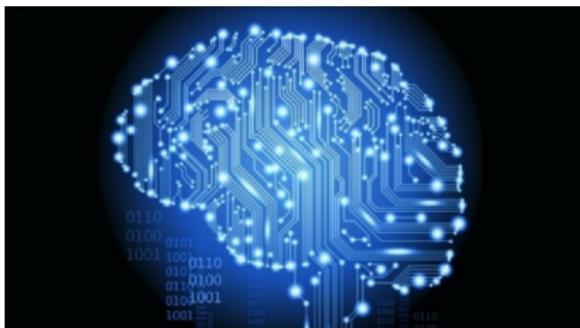
- Pourquoi une IA pour le Jeu de Go ?
- Le Jeu de Go

### 2 Avant l'Exploration d'Arbre par Bandit

### 3 Exploration d'Arbre par Bandit

### 4 Conclusion

## 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

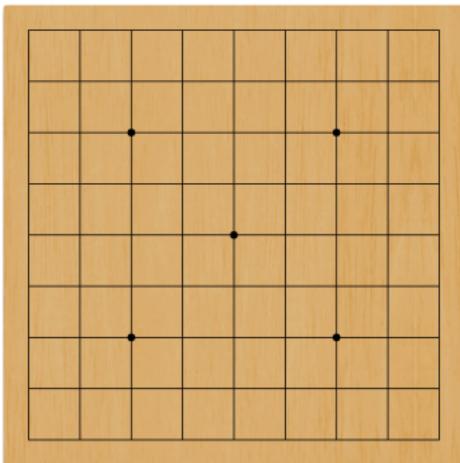
- plateau de jeu : Goban
- deux tailles 9x9 ou 19x19
- pierres noires et blanches





## Règles : placement

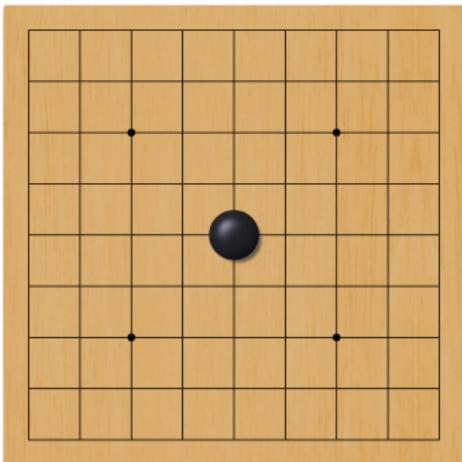
- Début de la partie : le plateau est vide
- Chaque joueur pose une pierre à tour de rôle
- Noir commence
- Pierres posées sur les intersections





## Règles : placement

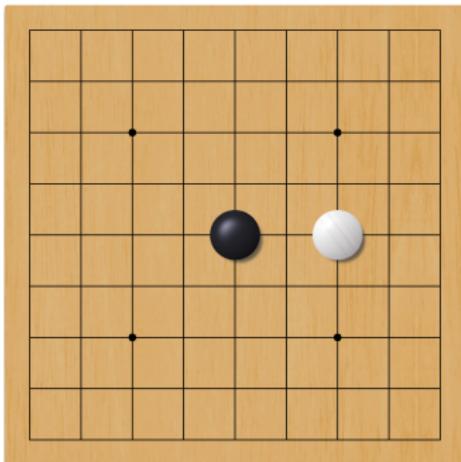
- Début de la partie : le plateau est vide
- Chaque joueur pose une pierre à tour de rôle
- Noir commence
- Pierres posées sur les intersections





## Règles : placement

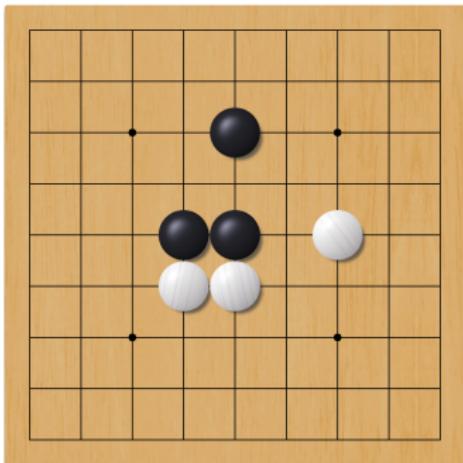
- Début de la partie : le plateau est vide
- Chaque joueur pose une pierre à tour de rôle
- Noir commence
- Pierres posées sur les intersections





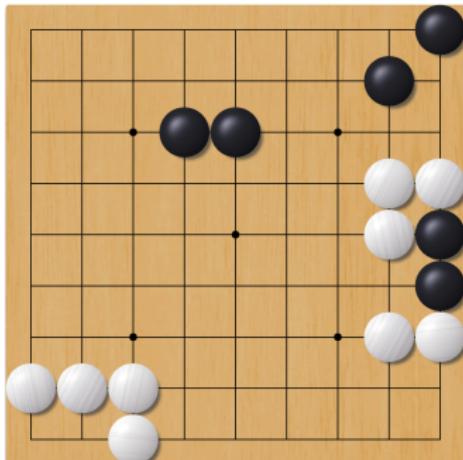
## Règles : placement

- 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

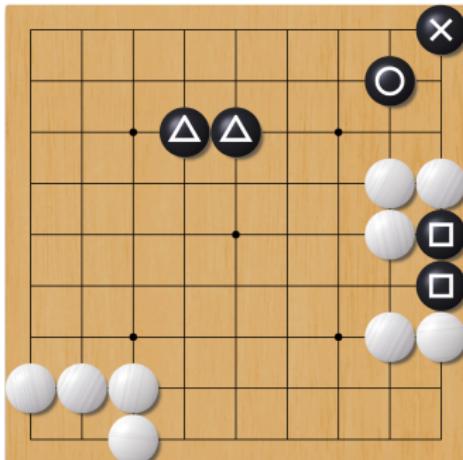
- Pierres reliées horizontalement ou verticalement : **une chaine**
- Emplacements libres autour d'une chaine : **liberté**





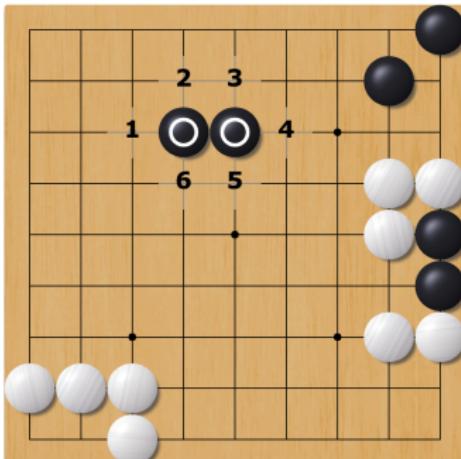
## Règles : chaînes et libertés

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



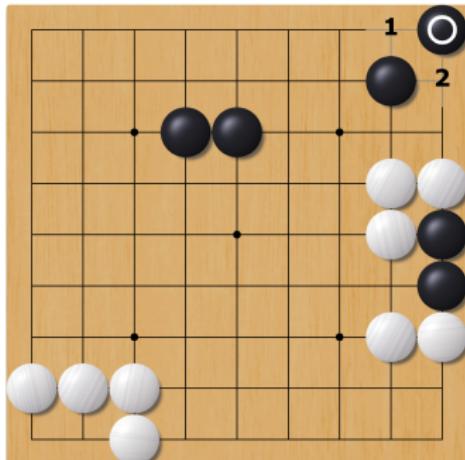
## Règles : chaînes et libertés

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



## Règles : chaînes et libertés

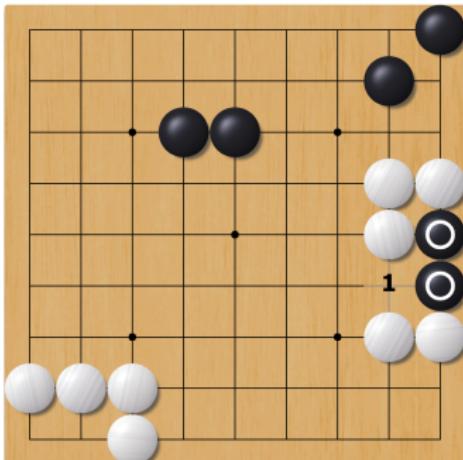
- Pierres reliées horizontalement ou verticalement : **une chaine**
- Emplacements libres autour d'une chaine : **liberté**





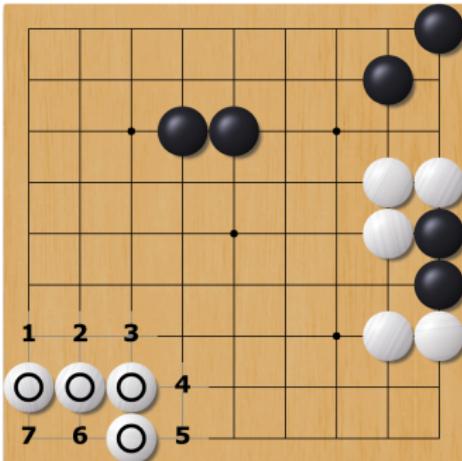
## Règles : chaînes et libertés

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



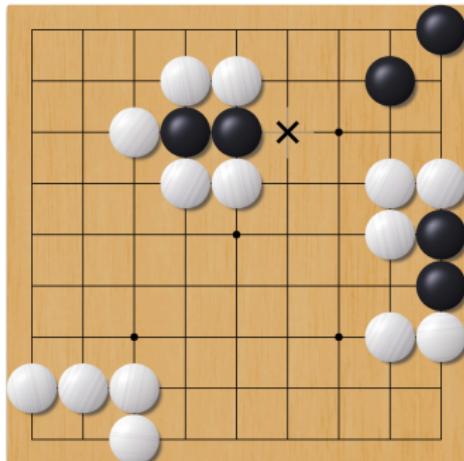
## Règles : chaînes et libertés

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



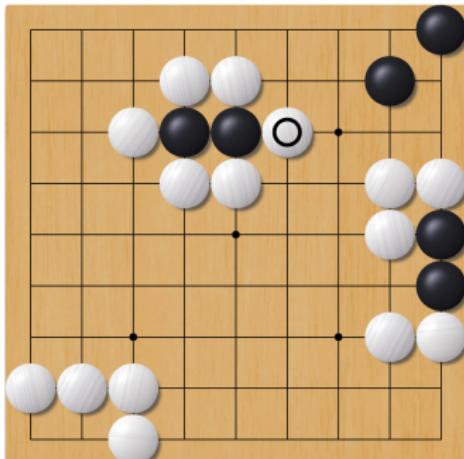
## Règles : capture

- Enlever la dernière liberté d'une chaîne : **capture**
- les pierres sont enlevées du plateau



## Règles : capture

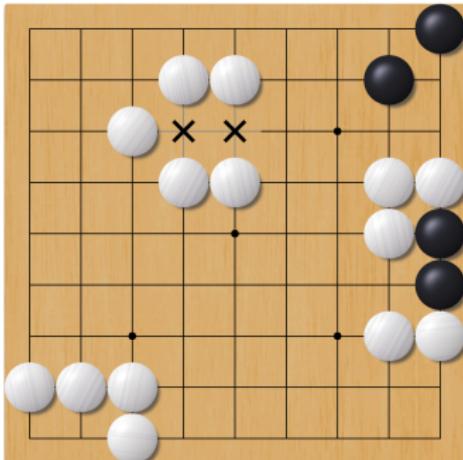
- Enlever la dernière liberté d'une chaîne : **capture**
- les pierres sont enlevées du plateau





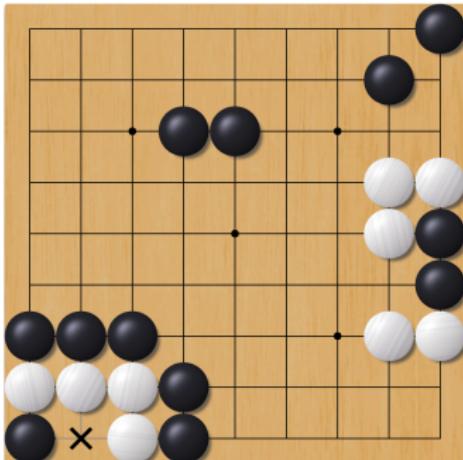
## Règles : capture

- Enlever la dernière liberté d'une chaîne : **capture**
- les pierres sont enlevées du plateau



## Règles : capture

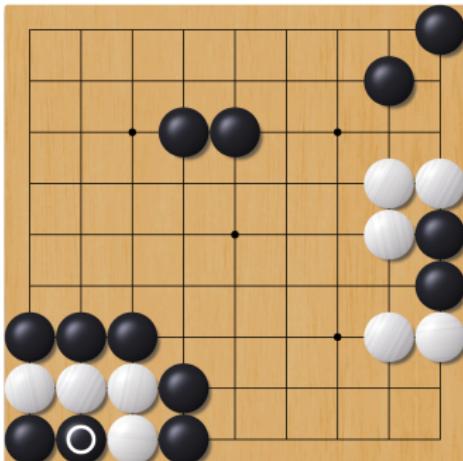
- Enlever la dernière liberté d'une chaîne : **capture**
- les pierres sont enlevées du plateau





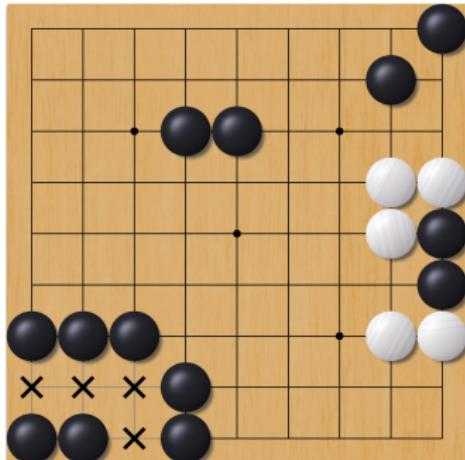
## Règles : capture

- Enlever la dernière liberté d'une chaîne : **capture**
- les pierres sont enlevées du plateau



## Règles : capture

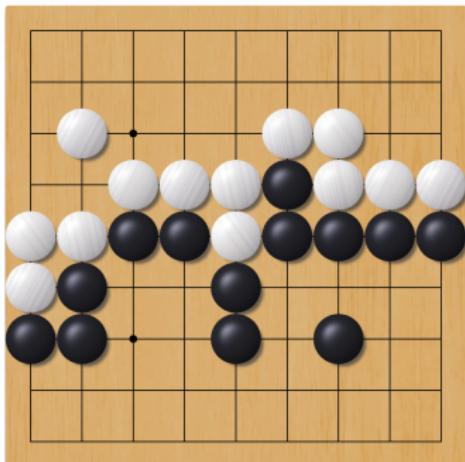
- Enlever la dernière liberté d'une chaîne : **capture**
- les pierres sont enlevées du plateau





## Règles : fin de partie

- partie terminée quand les deux joueurs passent
- score : pierres + territoire

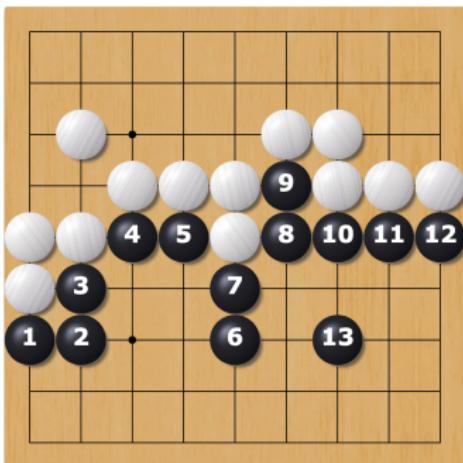


TODO score



## Règles : fin de partie

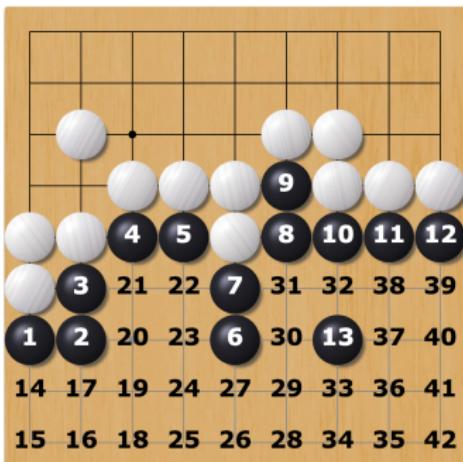
- partie terminée quand les deux joueurs passent
- score : pierres + territoire



TODO score

## Règles : fin de partie

- partie terminée quand les deux joueurs passent
- score : pierres + territoire

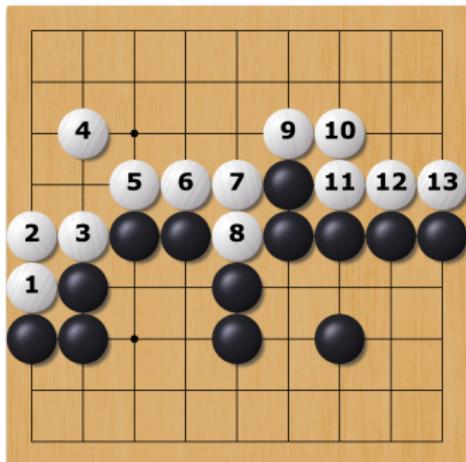


TODO score



## Règles : fin de partie

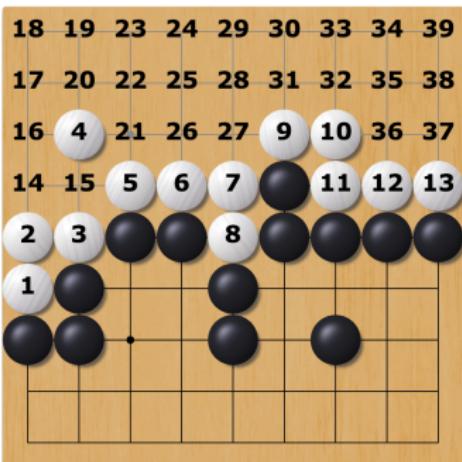
- partie terminée quand les deux joueurs passent
- score : pierres + territoire



TODO score

## Règles : fin de partie

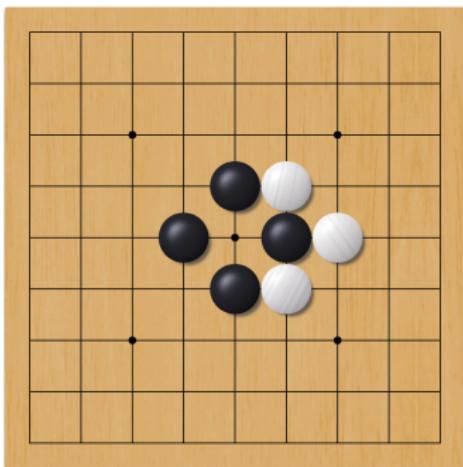
- partie terminée quand les deux joueurs passent
- score : pierres + territoire



TODO score



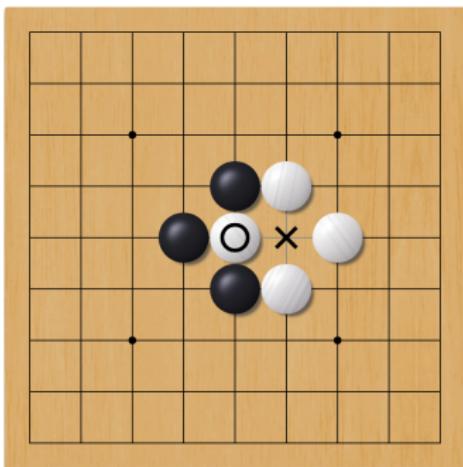
## Règles : le ko



- problème : captures répétées successives
- règle (humain) : pas le droit de remettre le plateau dans l'état juste avant
- règle (ordinateur) : pas le droit de remettre le plateau dans n'importe quel état précédent



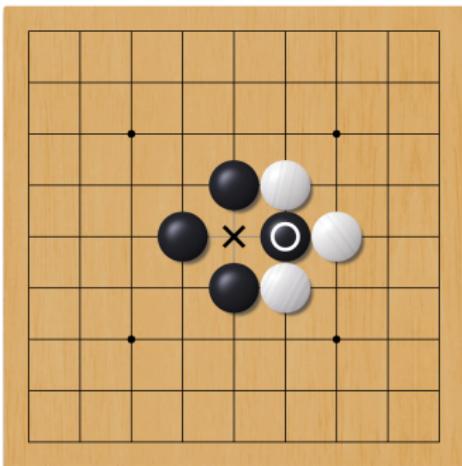
## Règles : le ko



- problème : captures répétées successives
- règle (humain) : pas le droit de remettre le plateau dans l'état juste avant
- règle (ordinateur) : pas le droit de remettre le plateau dans n'importe quel état précédent



## Règles : le ko



- problème : captures répétées successives
- règle (humain) : pas le droit de remettre le plateau dans l'état juste avant
- règle (ordinateur) : pas le droit de remettre le plateau dans n'importe quel état précédent



## Echelle de niveau

TODO tikz



## Plan

1 L'IA et le Jeu de Go

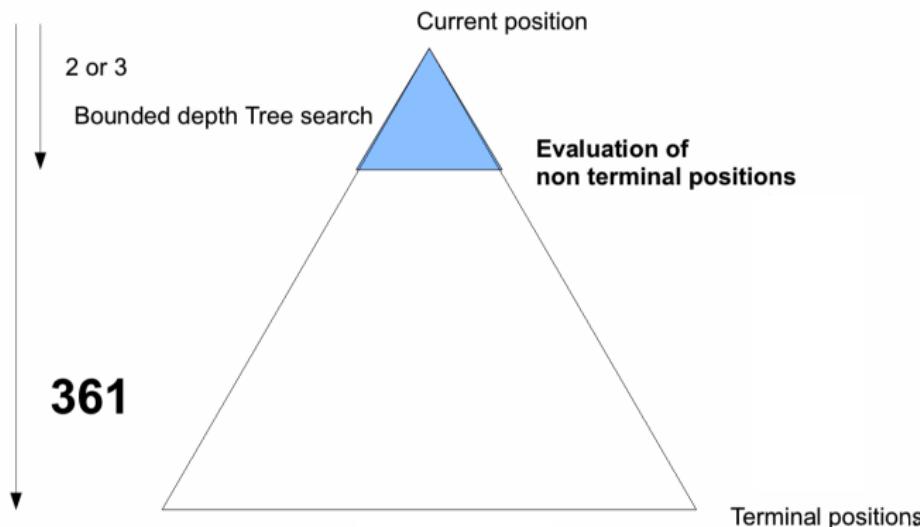
2 Avant l'Exploration d'Arbre par Bandit

3 Exploration d'Arbre par Bandit

4 Conclusion

## Principe

- Exploration d'arbre alphabeta
- Evaluation des noeuds basée sur des connaissances expertes



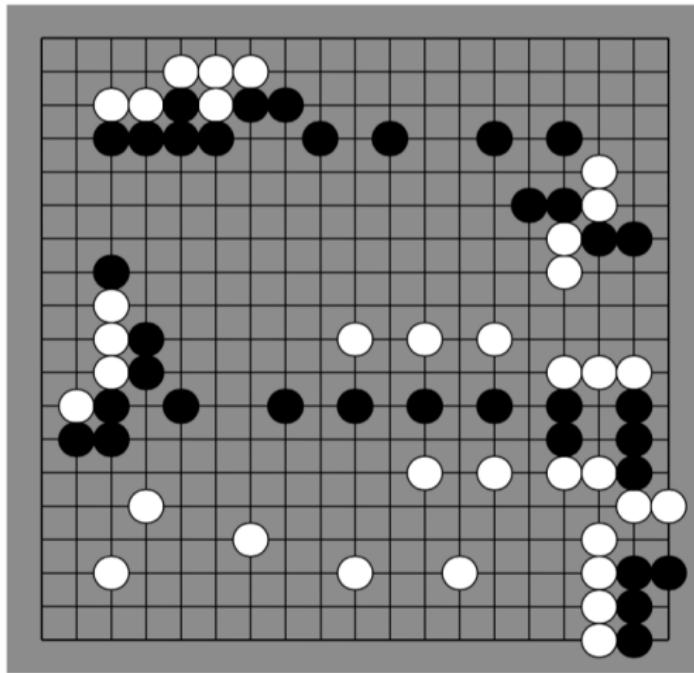


## Evaluation

- **Découpage** du plateau en sous parties
- **Evaluation** de chaque sous partie par **recherche locale**  
(souvent alphabeta)
- groupe mort, vivant, territoire, ...
- **Recomposition** d'un score global

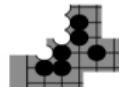
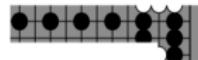
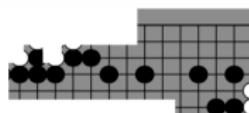


## Position à évaluer



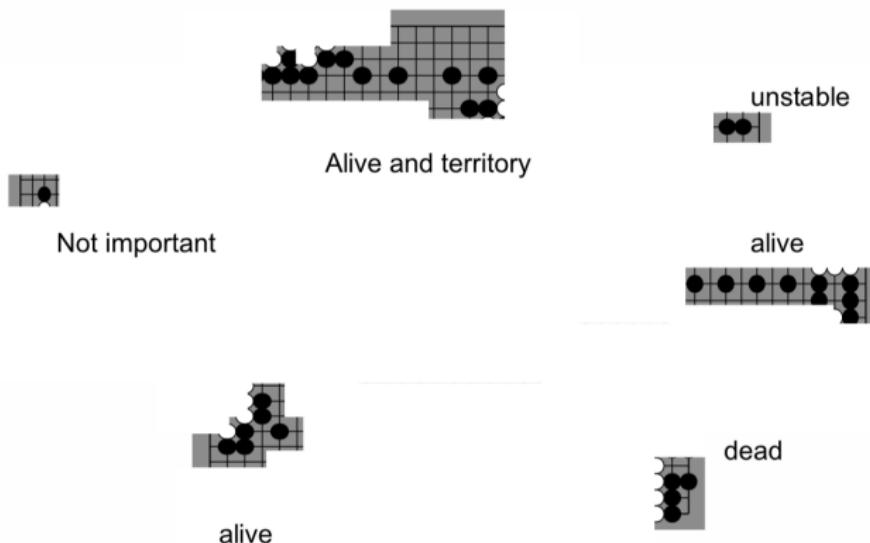


## Découpage du plateau



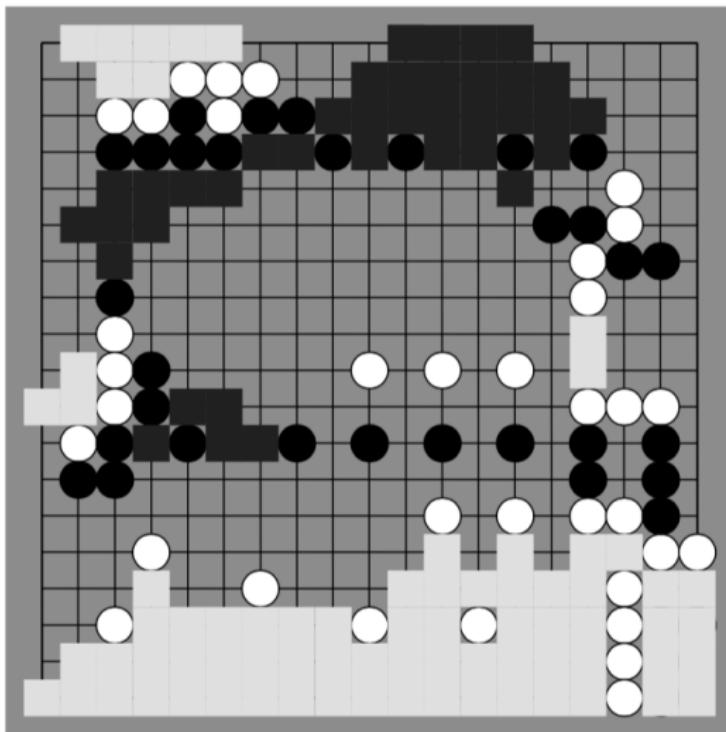


## Evaluation locale





## Evaluation globale





## Avantages

- Algorithme très rapide
- Evaluation locale peut être très performante



## Inconvénients

- Découpage et recomposition difficile et ayant un fort impact
- Pas d'**interaction** entre les positions locales
- Demande beaucoup de **connaissances 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
- 4 Conclusion



## Idée

- Arbre déséquilibré
- TODO
- Construction itérative



## Principe

- Répétition de ces 3 étapes : TODO



## Exemple

TODO tikz



## Questions

### TODO

- Comment faire l'évaluation ?
- Comment faire la descente ?

## 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 ?

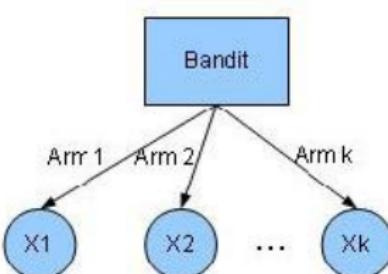
## Autres problèmes similaires



- 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  $\mu_k$ .
- 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 indépendant 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  $t$  :

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

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



## UCB

Borne sur le regret :

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

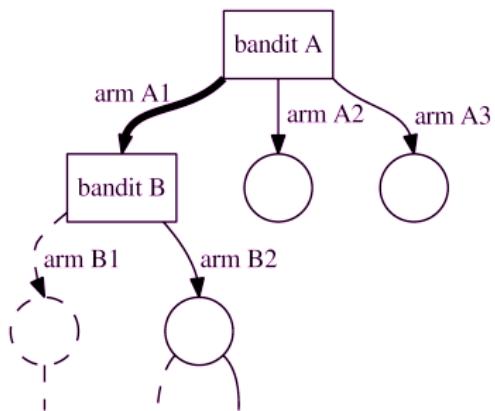


## Rappel MCTS

TODO rappel l'algo TODO rappel question : comment faire la descente ?

## 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  $C_i(t)$  :

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



## Améliorations

- Réduire le nombre de bras du bandit
- Ajout de connaissances expertes
- AMAF
- ...



## Echelle de niveau

TODO



## Première victoire en 9x9

TODO



## Plan

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



# AlphaGo

TODO principe en 1 slide



## Echelle de niveau

TODO



## Autres applications

TODO



## Conclusion

TODO



## References

TODO