4. Jeux

Pierre Karpman

Lycée Champollion MP Tronc Commun
https://membres-ljk.imag.fr/Pierre.Karpman/CPGE/2025/

Table des matières

Table des matières

Objet d'étude : jeux

On s'intéresse exclusivement aux jeux :

- à deux joueurs
- non coopératifs (il ne peut y avoir que ≤ 1 gagnant)
- tour par tour (sans contrainte temps-réel)
- à *information complète* (pas d'état caché pour un ou plusieurs joueurs)
  - aussi sans hasard (le hasard est un état caché : celui des résultats de tous les tirages à venir)

Exemples

- Échecs ; Go ; Shōgi ; Jeu de dames ; Othello
- Morpion ; Puissance 4 ; Awalé
- Jeu de Nim
- Hex

Contre-exemples

- Coinche (4 joueurs ; pas à information complète)
- Petits chevaux (même à deux joueurs : pas à information complète)
- Tout puzzle ; énigme (un joueur ou coopératif)

Objectifs : gagner

Développer un cadre et des algorithmes pour :

- Trouver des *stratégies gagnantes* (ou à défaut, de match nul)
  - mais ce n'est envisageable que pour des tout petits jeux
- Trouver des façons de jouer « pas trop bêtes »
  - en se basant sur une estimation de la qualité d'une position (une forme d'*heuristique*)

Table des matières

Jeu $\rightarrow$ graphe

Graphe d'un jeu

Un jeu (à deux joueurs etc.) peut se représenter par un graphe orienté $G = (S, A)$ où

- Les sommets S représentent les *positions* du jeu
- Les arcs A représentent les liens d'*accessibilité* entre positions

Addition pratique : on ajoute le tour du prochain joueur dans les positions : le graphe devient *biparti*

Graphe *biparti* d'un jeu

Le graphe d'un jeu devient de la forme $G = (S_0 \sqcup S_1, A)$, avec S_0, S_1, A t.qu'il n'existe aucun arc entre deux sommets de S_0 ou deux sommets de S_1

- G s'appelle l'*arène* du jeu
- S_0 (resp. S_1) sont les sommets contrôlés par le joueur 0 (resp. 1)

Partie

Sommets distingués

Soit $G = (S_0 \sqcup S_1, A)$ une arène, on distingue :

- un *sommet de départ* s_0 (conventionnellement $\in S_0$)
- des sommets de victoire T_0 (resp. T_1) pour le joueur 0 (resp. 1)
  - ce sont des *puits* du graphe : leur degré sortant est nul ; on suppose $T_0 \neq \varnothing$, $T_1 \neq \varnothing$
- éventuellement des sommets de match nul N
  - des puits qui ne sont ni dans T_0 ni dans T_1

Un jeu est modélisé par (G, s_0, T_0, T_1)

Partie

Une *partie* d'un jeu (G, s_0, T_0, T_1) est un chemin dans G d'origine s_0 ; elle est :

- une victoire pour le joueur 0 (resp. 1) si elle se termine en T_0 (resp. T_1)
- nulle si elle se termine en N ou si elle ne termine pas (le chemin est infini)
  - en pratique on ignorera la possibilité de parties infinies
- (partielle si le chemin est fini et termine en un sommet qui n'est pas un puits)

Stratégie

On considère des stratégies sans mémoire (ne dépendent pas des coups passés) et déterministes (n'utilisent pas de hasard)

- On note $S'_0 \subset S_0$, $S'_1 \subset S_1$ les sommets de S_0, S_1 qui ne sont pas des puits

Stratégie

Une stratégie (sans mémoire, déterministe) pour un jeu $(G = (S_0 \sqcup S_1, A), -, -, -)$ et le joueur i est une application $\sigma : S'_i \to S_{1-i}$ t.q. $\forall\, s \in S'_i, \sigma(s) = t \Rightarrow (s,t) \in A$: pour toute position non terminale, σ indique un (unique) coup légal à jouer

- Une partie (un chemin dans G) est *jouée suivant une stratégie* σ pour le joueur i si le chemin s'écrit $\cdots \to s^i_k \to \sigma(s^i_k) \to s^i_{k+1} \to \sigma(s^i_{k+1}) \to \cdots$ où les $s^i_j \in S_i$: pour tout sommet de S_i visité, le prochain sommet visité est donné par σ
- Une stratégie σ est *gagnante* pour i ssi. toute partie jouée par i suivant σ termine en T_i (et ce *pour toute stratégie* de $1-i$)

Table des matières

Attracteur

Position gagnante

Une *position gagnante* pour i dans un jeu $((S_0 \sqcup S_1, A), s_0, T_0, T_1)$ est un sommet w t.qu'il existe une stratégie gagnante pour i à partir de w

- i est garanti de gagner à partir de w en jouant « au mieux »
- tout sommet de T_i est une position gagnante pour i
  - donc on n'est pas en train de définir quelque chose qui ne peut pas exister…

Attracteur

L'ensemble des positions gagnantes pour un joueur i est appelé *attracteur* de i, noté $\mathfrak{A}^i$

- 0 dispose d'une stratégie gagnante pour *tout* le jeu ssi. $s_0 \in \mathfrak{A}^0$

Calcul des attracteurs

Le calcul des attracteurs (sans parties infinies) se fait « aisément » par récurrence

Attracteur : définition récursive

- On note $\mathfrak{A}_t^i$ les positions gagnantes w de i tel qu'il existe un chemin $w \rightsquigarrow T_i$ de longueur *au plus* $2t$ (nécessite au plus t coups pour i)
- $\mathfrak{A}^i = \bigcup_{t \geq 0} \mathfrak{A}_t^i$
  - $= \bigcup_{t \leq N} \mathfrak{A}_t^i = \mathfrak{A}_N^i$ pour un certain N par hypothèse d'absence de parties infinies
- $\mathfrak{A}_0^i = T_i$
- w est dans $\mathfrak{A}_t^i$ si :
  - il est une position gagnante en $\leq t$ coups pour i, donc il est dans $\mathfrak{A}_{t-1}^i$ ou :
  - il existe **un** $w' \in S_{1-i}$ accessible depuis w t.q. **tous** les $w'' \in S_i$ accessibles depuis w' sont dans $\mathfrak{A}_{t-1}^i$

Attracteur : calcul « en remontant »

Idée : on initialise $\mathfrak{A}_0^i$ à T_i puis tant que $\mathfrak{A}_t^i \neq \mathfrak{A}_{t-1}^i$:

- pour chaque prédécesseur w' de $w \in \mathfrak{A}_t^i$: si tous les successeurs de w' sont dans $\mathfrak{A}_t^i$, on ajoute tous les prédécesseurs de w' à $\mathfrak{A}_{t+1}^i$

Attracteur : calcul alternatif

On peut calculer si une position donnée est dans un attracteur « en descendant »

- On suppose l'arène (finie) acyclique (sinon il faut ajouter une détection de cycle)
- On étend la définition des attracteurs : $u \in S_i$ dans $\mathfrak{A}^{1-i}$ si quand i joue depuis u il ne peut **que** perdre si $1-i$ joue au mieux

Attracteur : calcul « en descendant »

Une position $u \in S_i$ est :

- dans $\mathfrak{A}^{1-i}$ si **tous** ses successeurs sont dans $\mathfrak{A}^{1-i}$
- dans $\mathfrak{A}^{i}$ si l'**un** de ses successeurs est dans $\mathfrak{A}^{i}$
- dans aucun attracteur sinon

Calcul récursif possible, avec T_0, T_1 et N comme cas de base ; terminaison garantie !

Programmation dynamique !

- Le calcul en remontant est efficace (mais peut calculer plus que nécessaire)
- Le calcul en descendant peut être inefficace en cas de sous-cas non disjoints traités plusieurs fois
  - Solution : mémoïsation !

Attracteurs : commentaires

- Tout jeu peut être complètement « résolu » en calculant $\mathfrak{A}^0$ et $\mathfrak{A}^1$
  - Ou bien s_0 est dans un attracteur et un joueur jouant parfaitement peut toujours gagner, ou bien deux joueurs jouant parfaitement peuvent toujours faire match nul (ex. : morpion : https://xkcd.com/832 ; jeu de Nim)
- Le calcul des attracteurs se fait efficacement (typiquement linéairement) en la taille de l'arène
- Mais la taille de l'arène est généralement gigantesque !
  - Go : $\lessapprox 10^{170}$
  - Échecs : $\lessapprox 10^{46}$
  - Puissance 4 : $\approx 10^{11.6}$
- Méthode peu applicable en pratique

Table des matières

Attracteur version MinMax

- On associe une valeur aux puits du graphe : $+\infty$, $-\infty$, 0 pour ceux de T_0, T_1, N respectivement
- On peut redéfinir l'objectif du joueur 0 « Max » (resp. joueur 1 « Min ») comme terminer en un sommet de valeur max (resp. min)

Attracteur : calcul « MinMax »

Le calcul descendant des attracteurs version MinMax devient :

- Pour Max : la valeur d'un sommet $u \in S_{\max} := \max\{\text{valeur des successeurs de } u\}$
- Pour Min : la valeur d'un sommet $u \in S_{\min} := \min\{\text{valeur des successeurs de } u\}$

Intérêt : exploration non exhaustive

- On peut arrêter la récursion avant les cas de base
- Mais il faut alors déterminer la valeur d'une position non terminale (pour en faire un nouveau cas de base)
- Comme on sait pas, on devine (et on parle d'« heuristique » pour faire bien)

MinMax avec heuristique

Exemple : échecs

- Heuristique : somme pondérée des pièces blanches (Max) moins celle des pièces noires (Min) encore en jeu (exemple de pondération « classique » (Tartacover) : ♕ = 10 ; ♖ = 5 ; ♘ = ♗ = 3¼ ; ♙ = 1)
- Ignore toute notion de position : peu d'intérêt à profondeur 1, mais pas absurde à profondeur raisonnable

Implémentation

Plusieurs types d'approche / compromis :

- profondeur fixe ou « budget » fixe (quand le nombre de cas à traiter pour une profondeur est variable)
- compromis entre profondeur et sophistication de l'heuristique

On peut facilement éviter certains appels récursifs 🙈