Intelligence Artificielle et Algorithmique

Fabien Torre

université de Lille

6 mars 2019

Une (de plus !) définition de l'Intelligence Artificielle

Jean-Louis Laurière (1986)

L'IA commence là où l'informatique classique s'arrête :
tout problème pour lequel il n'existe pas d'algorithme connu
ou raisonnable permettant de le résoudre relève *a priori* de l'IA.

Interprétations algorithmiques

- pas d'algorithme : le problème est indécidable.
- pas d'algorithme raisonnable : tout algorithme résolvant le problème est de complexité exponentielle.

Explosion exponentielle : les morts-vivants

Un enseignant vient de muter en une créature indéterminée.

Il est pour l'instant le seul de son espèce mais il va chaque jour, dès demain, contaminer et transformer une nouvelle personne.

À son tour, chaque nouveau muté va chaque jour, à partir du lendemain de sa propre transformation, contaminer une personne.

Combien de temps sera nécessaire pour infecter toute l'humanité (soit 7 milliards et 500 millions de personnes) ?

34 jours !

Algorithme non raisonnable : les tours de Hanoï

(a) (b) (c)

Objectif et règles du jeu

- déplacer la pyramide du piquet (a) au piquet (c),
- en bougeant un plateau à la fois,
- sans jamais poser un plateau sur plus petit que lui.

... et avec plus de plateaux ? n plateaux nécessitent $2^n - 1$ actions.

Loi de Moore et angle d'attaque

La loi de Moore... n'est d'aucune aide !
Comment en IA attaque-t-on des problèmes exponentiels ?

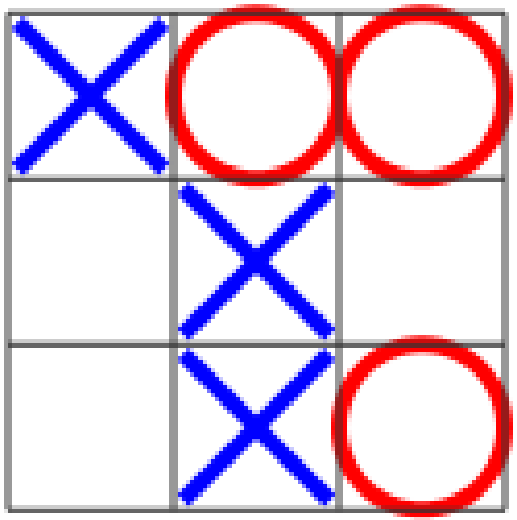
1. Jeux à deux joueurs (morpion, puissance 4, reversi, etc.)
2. Casse-têtes (compte est bon, taquin, labyrinthes, etc.)

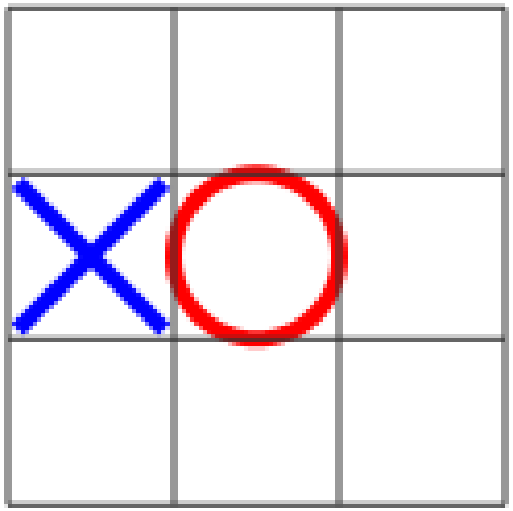
Pistes

- partir d'algorithmes classiques,
- injecter de la connaissance experte,
- évaluer expérimentalement, par compétition.

L'exemple du morpion

Dans ces deux situations, *comment décider du coup à jouer ?*

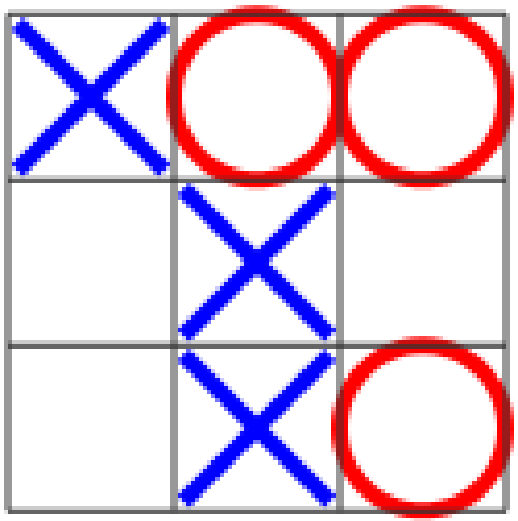




Fonction heuristique

Heuristique (du grec εὑρίσκω, eurisko, *je trouve*)

- [psychologie] opération (mentale), rapide et intuitive.
- « une méthode empirique de résolution de problème, dont la validité ou l'efficacité n'est pas prouvée »,
- dans la vie réelle ? choisir une file d'attente, etc.



- 2 lignes possibles pour les ✕
- 1 ligne possible pour les ○
- l'état vaut 2-1 = +1 pour les ✕

Importance et caractéristiques de la fonction heuristique

Succès = qualité de la heuristique + descente en profondeur.

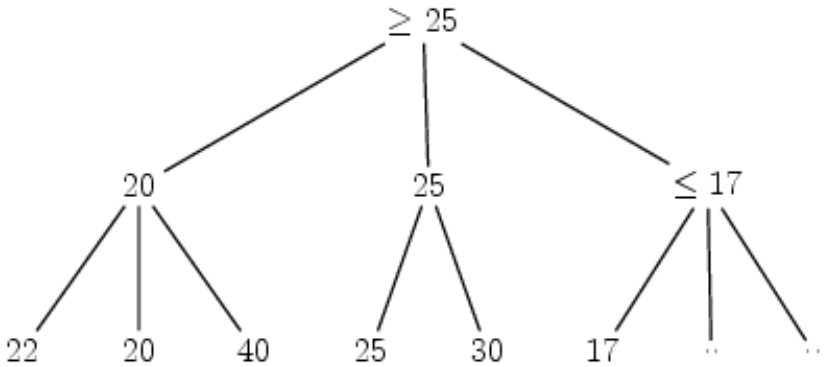
La clef c'est donc la heuristique qui idéalement doit être...

- rapide à calculer pour permettre une exploration profonde,
- nulle pour les matches nuls,
- positive si la situation est favorable au joueur artificiel,
- négative sinon,
- diverse selon les états,
- pertinente !

Permet à chacun d'exprimer sa vision du jeu !

Une optimisation possible : les coupes

Sur un arbre dont la racine est de type MAX :



On peut couper sans risque sous le nœud ≤ 17.

Coupe sous un nœud MIN : coupe alpha.

Les jeux en IA

Domaine actif de l'Intelligence Artificielle depuis sa naissance...

- dames anglaises (damier 8×8) : de 1947 à 2007 (résolu),
- échecs : de 1967 à 1997 (champion du monde),
- go : de 1974 à 2017 (champion du monde).

Autres jeux possibles : jeu de Nim, morpion, puissance 4, reversi, awalé, dames, échecs, go, *breakthrough*, Havannah, Hex, etc.

Méthodes vues

1. arbre complet et remontée des fins de partie par min-max,
2. arbre limité, remontée par min-max des fins de partie et 0 si partie en cours,
3. arbre limité, remontée par min-max des fins de partie et valeur heuristique si partie en cours,
4. idem avec coupes : algorithme alpha-bêta.

Selon la difficulté du jeu et les niveau/motivation des étudiants !

Implémentation

Algorithmes génériques à instancier par quelques méthodes spécifiques au jeu considéré.

Pour chaque jeu

- codage des états du jeu et des coups,
- `fin_partie : état -> booléen`
- `coups_possibles : état x joueur -> liste de coups`
- `applique_coup : état x coup -> état`
- `heuristique : état -> nombre réel`

1. Jeux à deux joueurs (morpion, puissance 4, reversi, etc.)
2. Casse-têtes (compte est bon, taquin, labyrinthes, etc.)
3. Bilan

Exemple : le problème des cruches

Vous disposez de deux cruches, l'une de 7 litres, l'autre de 5 litres.
Une source d'eau est à disposition, à volonté.
Comment obtenir exactement 4 litres ?

un état = un couple (grande cruche, petite cruche)
état initial : $(0, 0)$
états finaux : $(4, ?)$ et $(?, 4)$

opérateurs : RemplirPetite RemplirGrande JeterPetite JeterGrande ViderPetiteDansGrande ViderGrandeDansPetite CompléterPetite CompléterGrande

Enchaînement d'opérateurs amenant de l'état initial à l'état final ?

Exemple : le taquin

état initial

2	8	3
1	6	4
7		5

état final

1	2	3
8		4
7	6	5

un état = une matrice 3×3 + les coordonnées de la case vide

opérateurs (qui déplacent la case vide !) :
`HAUT`, `BAS`, `GAUCHE` et `DROITE`.

Enchaînement d'opérateurs amenant de l'état initial à l'état final ?

Bilan des recherches non informées

Sur le taquin

- $9! = 362\,880$ états différents,
- 181 440 réellement accessibles depuis un état initial fixé,
- facteur de branchement ≥ 2, profondeur 30.

Sur les méthodes

- méthodes trop gourmandes en mémoire ou en temps,
- sans garantie de trouver la solution la plus courte.

Objectifs

- accélérer la recherche,
- limiter l'occupation mémoire,
- garantir la terminaison et la solution optimale.

A^* : fonctions d'évaluation

Principes

- *information heuristique* pour orienter la recherche,
- algorithme A^* pour l'utiliser.

Fonctions

Nous nous donnons une fonction f capable d'évaluer un nœud n :

$$f(n) = g(n) + h(n)$$

- $g(n)$ coût du meilleur chemin connu pour atteindre n,
- $h(n)$ estimation du coût du chemin entre n et un état final,
- $f(n)$ estimation du coût de la meilleure solution passant par n.

Heuristique pour le taquin

Il s'agit d'évaluer le nombre d'étapes nécessaires entre l'état courant et l'état final...

état courant

6	4	3
8		5
2	1	7

état final

1	2	3
8		4
7	6	5

On peut utiliser la distance de Manhattan...

$$h(\text{état courant}) = 3 + 3 + 0 + 2 + 1 + 3 + 2 + 0 = 14$$

Importance et caractéristiques de la fonction heuristique

La clef c'est donc la fonction h qui idéalement doit être...

- nulle pour les états finaux,
- rapide à calculer,
- proche de h^*,
- mais minorante : $\forall n : h(n) \leq h^*(n)$,
- diverse selon les états,
- monotone.

Méthodes vues

1. recherche en profondeur,
2. recherche en profondeur limitée,
3. recherche en profondeur avec mémoire,
4. recherche en largeur,
5. algorithme A^* (utilisation d'une heuristique).

Discussion sur chaque méthode

termine toujours ou pas ? trouve une solution s'il en existe ?
trouve la solution la plus courte ?
occupation en mémoire ? temps de calcul ?

Implémentation

Algorithmes génériques à instancier par quelques méthodes spécifiques au problème considéré.

Pour chaque problème

- codage des états et des opérateurs,
- `état_initial : état`
- `est_final(e) : état -> booléen`
- `opérateurs_disponibles : liste d'opérateurs`
- `heuristique : état -> nombre réel`

Autres casse-têtes possibles : « le compte est bon », sokoban, labyrinthes, rubik's cube, etc.

En Intelligence Artificelle

Rappels sur le test de Turing

- imiter le comportement, pas le fonctionnement,
- dialoguer avec les experts humains.

Les méthodes vues

- ne fonctionnent pas comme l'esprit humain,
- peuvent expliquer leurs décisions,
- battent leur programmeur et de très bons joueurs,
- se basent sur des heuristiques.

Intérêts pour l'enseignement de l'informatique

Nous avons vu deux algorithmes génériques qui :

- outrepassent la nature exponentielle du problème, par l'utilisation d'heuristiques,
- laissent la main sur la définition des heuristiques,
- nécessitent de bien modéliser les problèmes à résoudre, *a priori* très différents,
- explorent des graphes sous-jacents, ramenés à des arbres, que l'on essaye de ne pas développer complètement,
- permettent variantes/discussion sur les propriétés de chacune,
- demandent une implémentation générique et efficace, programmation séquentielle ou récursive,
- ouvrent la possibilité de programmer des interfaces.

Ressources

1. Cours en Intelligence Artificielle
2. Exercices en Intelligence Artificielle