

IA dans les Jeux : Concepts de Base et Fondements

Notes de cours 2

Nadjib Lazaar
Université Paris-Saclay
lazaar@lisn.fr

Pourquoi les jeux ?

Les jeux constituent un domaine privilégié pour l'IA, car ils offrent un cadre formel, structuré et mesurable permettant d'évaluer l'intelligence et l'efficacité des algorithmes. En modélisant des environnements compétitifs aux règles précises, les jeux fournissent un terrain d'expérimentation idéal pour développer des agents capables de planifier et d'agir de manière optimale. Ces agents doivent analyser les situations, anticiper les actions adverses, et prendre des décisions qui maximisent leurs chances de succès, en s'appuyant sur des stratégies adaptées et des techniques algorithmiques avancées.

1 Jeu et Agent Intelligent

Un **jeu à deux joueurs** est un jeu compétitif où deux joueurs alternent leurs actions à chaque tour. Chaque joueur cherche à maximiser son propre gain tout en minimisant celui de l'autre. Ces jeux sont souvent **à somme nulle**, ce qui signifie que le gain d'un joueur est exactement la perte de l'autre. En d'autres termes, la somme des gains des deux joueurs est constante et égale à zéro à chaque état final du jeu.

Jeu à Deux Joueurs

- Un jeu \mathcal{J} à deux joueurs est défini par un quintuple $\langle \mathcal{P}, S, A, T, U \rangle$, où :
- $\mathcal{P} = \{p_1, p_2\}$: L'ensemble des **joueurs**, où chaque joueur a un objectif opposé à celui de l'autre (maximisation de l'utilité pour chaque joueur).
 - S : L'ensemble des **états** du jeu, représentant toutes les configurations possibles du système de jeu.
 - A : L'ensemble des **actions** disponibles pour chaque joueur. Chaque joueur choisit une action à chaque tour, en fonction de l'état actuel.
 - $T : S \times A \rightarrow S$: La fonction de **transition**, qui détermine l'état suivant en fonction de l'état actuel et de l'action choisie par chaque joueur.
 - $U : S \rightarrow \mathbb{R}^2$: La fonction **d'utilité**, qui associe à chaque état terminal $s \in S$ un couple de valeurs (u_1, u_2) , représentant les gains des joueurs p_1 et p_2 , respectivement.

Dans ce type de jeu, l'objectif de chaque joueur est de maximiser sa propre utilité $U_1(s)$ pour p_1 , ou $U_2(s)$ pour p_2 , tout en tenant compte des actions de l'adversaire. Les joueurs jouent de manière alternée : le joueur p_1 choisit une action à chaque tour, puis p_2 réagit, et ainsi de suite jusqu'à ce que le jeu atteigne un état terminal.

Un **agent intelligent** est une entité autonome qui perçoit son environnement et prend des décisions pour atteindre un objectif donné, en fonction de ses perceptions.

Agent Intelligent

Soit un jeu $\mathcal{J} = \langle \mathcal{P}, S, A, T, U \rangle$, où L'agent \mathcal{A} utilise une fonction de **perception** $f : S \rightarrow O$, qui lui permet de percevoir son environnement et d'obtenir des informations sur l'état actuel s . La fonction de **récompense** $r : S \rightarrow \mathbb{R}$ associe une valeur à chaque état s , représentant la qualité de cet état pour l'agent.

L'agent choisit ses actions $a \in A$ en fonction de ses perceptions $o \in O$ afin de maximiser une fonction d'utilité ou de récompense sur un horizon temporel donné. Cette action est déterminée par une politique $\pi : O^* \rightarrow A$, qui spécifie l'action à choisir en fonction des observations passées.

Dans un jeu compétitif, l'agent intelligent cherche à maximiser son propre gain tout en tenant compte des actions de l'adversaire. L'agent adopte une stratégie qui vise à se rapprocher de l'**oracle** — c'est-à-dire la stratégie optimale qui maximiserait sa récompense en supposant qu'il connaît parfaitement l'environnement et les actions de l'adversaire. L'agent ajuste ses choix en fonction de l'historique de ses perceptions, en tentant d'imiter ou d'approcher le comportement optimal dans le jeu.

La performance de l'agent est mesurée par la somme des récompenses reçues sur une séquence d'états s_0, s_1, \dots, s_t :

$$\max_{\pi} \mathbb{E} \left[\sum_{t=0}^T r(s_t) \right]$$

où \mathbb{E} représente l'espérance mathématique et T est l'horizon de planification. L'agent peut également utiliser des stratégies d'apprentissage pour se rapprocher de la stratégie optimale, en ajustant ses actions au fil du temps en fonction de l'expérience accumulée.

2 L'Arbre de Jeu

Un **arbre de jeu** est une structure qui modélise toutes les configurations possibles d'un jeu et les décisions des joueurs à chaque étape. Dans un cadre de jeu à deux joueurs, l'arbre de jeu représente les alternances entre les actions des joueurs, en partant d'un état initial et en explorant toutes les séquences possibles de coups jusqu'à un état terminal.

Arbre de Jeu

Soit un jeu à deux joueurs $\mathcal{J} = \langle \mathcal{P}, S, A, T, U \rangle$. L'arbre de jeu associé est une structure arborescente définie comme suit :

- Les **nœuds** de l'arbre représentent les **états** du jeu $s \in S$,
- Les **arcs** entre les nœuds représentent les **actions** $a \in A$ effectuées par les joueurs,
- La **racine** de l'arbre correspond à l'état initial s_0 ,
- Les **feuilles** de l'arbre représentent les **états terminaux** où le jeu est terminé.

Exemple : Placement de Dominos sur un Tableau $n \times n$

Considérons un tableau $n \times n$ où les joueurs placent à tour de rôle un domino : l'un les place horizontalement, tandis que l'autre les place verticalement. Chaque domino occupe exactement deux cases adjacentes. Le jeu se termine lorsque plus aucun domino ne peut être placé. Le joueur qui effectue le dernier placement gagne la partie. La Figure 1 représente l'arbre de jeu d'un tableau 3×3 .

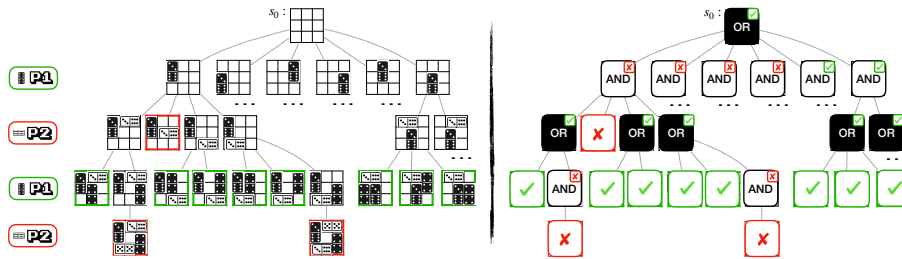


FIGURE 1 – Arbre de jeu des dominos (3×3). À gauche, on a l'arbre de jeu, tandis qu'à droite, il montre l'arbre AND/OR. Notez que les nœuds avec les trois petits points en bas sont **symétriques** aux nœuds déjà développés. Les nœuds AND correspondent aux décisions du joueur 1, qui cherche à garantir une victoire en maximisant ses options. Les nœuds OR représentent les décisions du joueur 2, qui vise à minimiser les chances de victoire du joueur 1. Les feuilles de l'arbre sont étiquetées "Victoire" ou "Défaite" selon l'issue du jeu pour le joueur 1.

Question : Trouver la Stratégie Gagnante

Après avoir exploré l'arbre de jeu complet pour le placement de dominos sur un tableau, réfléchissez à la stratégie optimale pour le joueur qui commence.

1. En consultant l'arbre de jeu complet, quelle stratégie permet au joueur 1 de garantir une victoire, quel que soit le comportement du joueur 2 ?
2. Identifiez les configurations critiques où un mauvais choix pourrait conduire à une défaite.

Cette réflexion permettra de mieux comprendre comment un agent intelligent peut analyser un arbre de jeu et adopter une stratégie optimale en fonction des configurations possibles.

2.1 Stratégie Gagnante pour le Joueur 1

Dans un arbre de jeu, chaque décision du joueur influence l'état futur du jeu. Pour déterminer la stratégie gagnante, un AND/OR arbre est particulièrement utile, car il modélise les décisions des deux joueurs, où :

- Les nœuds AND correspondent aux décisions du joueur 1, qui cherche à maximiser son gain.
- Les nœuds OR correspondent aux décisions du joueur 2, qui cherche à minimiser le gain du joueur 1.

L'AND/OR arbre représente une alternance entre ces deux types de nœuds, et permet de déterminer la meilleure stratégie à adopter par chaque joueur.

Voici comment déterminer la stratégie gagnante en suivant cet arbre de jeu à travers les étapes suivantes :

1. **Analysez les feuilles** : Les feuilles de l'arbre représentent les états terminaux du jeu. Chaque feuille est étiquetée "*Joueur-1*" (victoire du joueur 1) ou "*Joueur-2*" (victoire du joueur 2), selon le résultat du jeu. Ce sont ces états finaux qui vont déterminer la victoire ou la défaite.
2. **Remontez les valeurs des nœuds** :
 - Pour un **nœud OR**, qui représente le choix du joueur 2, on suppose que ce joueur cherche à minimiser les chances du joueur 1. Il choisira la branche avec la valeur minimale (c'est-à-dire, la branche qui mène à une "Défaite" pour le joueur 1 si possible).
 - Pour un **nœud AND**, qui représente le choix du joueur 1, ce joueur cherche à maximiser ses chances de victoire. Il choisira la branche qui mène à la "Victoire" si elle est disponible, et il garantira que toutes les branches mènent à des résultats favorables pour lui.
3. **Racine (nœud AND)** : En remontant l'arbre, on arrive au nœud racine, qui représente l'état initial du jeu. Si ce nœud peut être évalué à "Victoire" en tenant compte des choix de chaque joueur et des branches disponibles, alors le joueur 1 a une stratégie gagnante. Si la racine est évaluée à "Défaite", alors le joueur 2 peut forcer une défaite pour le joueur 1, quelle que soit la stratégie de ce dernier. Figure 1 illustre l'arbre AND/OR pour le jeu du domino.

2.2 De la Stratégie Gagnante à une Politique à Suivre

La stratégie gagnante, déterminée à partir de l'arbre de jeu, représente l'ensemble des décisions optimales permettant au joueur 1 de garantir une victoire, quelles que soient les réponses de l'adversaire. Pour exploiter cette stratégie de manière pratique, elle est transformée en une **politique de jeu**, qui associe chaque état s à l'action optimale a .

Une politique de jeu agit comme une table de correspondance ou une règle simple à suivre :

- À chaque tour, le joueur observe l'état actuel du jeu.
- Il consulte la politique pour déterminer l'action optimale en fonction de cet état.
- Il exécute cette action, tout en anticipant les contre-mouvements de l'adversaire.

La politique garantit que le joueur reste sur les chemins favorables identifiés dans l'arbre. En cas d'erreur adverse (l'adversaire choisit une action non optimale), la politique peut être ajustée localement pour exploiter les nouvelles opportunités. Ainsi, la stratégie gagnante se traduit en un plan clair et exploitable pour maximiser les chances de victoire.

2.3 Complexité des Arbres de Jeu

Les arbres de jeu souffrent de l'explosion combinatoire, une augmentation exponentielle de la taille de l'arbre à mesure que le facteur de branchement et la profondeur augmentent. En général, la taille de l'arbre dépend de deux paramètres principaux :

- n , le **facteur de branchement**, qui correspond au nombre de choix possibles (actions) à partir d'un état donné.
- d , la **profondeur**, qui correspond au nombre maximum de niveaux dans l'arbre. Cela est souvent lié au nombre total d'actions à effectuer avant d'atteindre un état terminal.

La taille de l'arbre dans le pire des cas est donnée par la complexité suivante :

$$O(n^d)$$

Ce comportement exponentiel rend rapidement l'exploration exhaustive impossible à mesure que n ou d augmente.

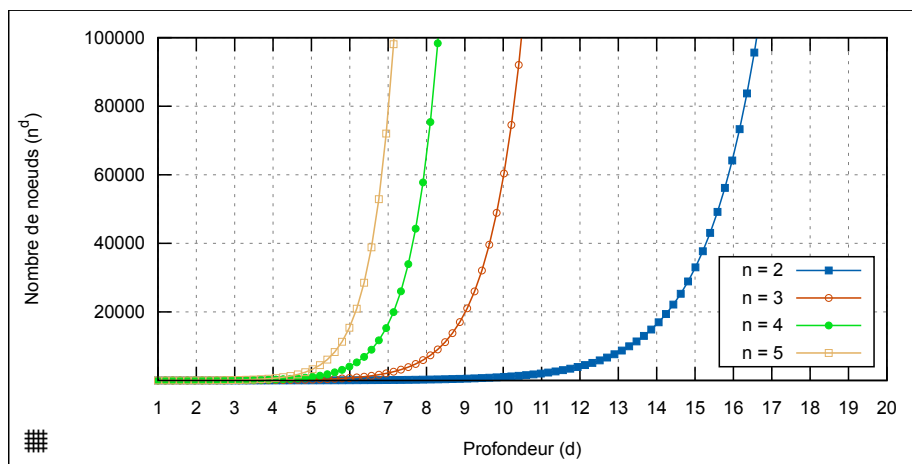


FIGURE 2 – Explosion combinatoire dans un arbre de jeu avec $n \in [2, 5]$.

Réduction de la Complexité

Pour limiter l'explosion combinatoire, des techniques d'optimisation comme l'élagage Alpha-Bêta sont utilisées. Ces méthodes permettent de :

- Ignorer les branches inutiles qui ne peuvent pas influencer la décision finale.
- Réduire la profondeur effective en appliquant des heuristiques ou des limites sur la recherche.

Ainsi, bien que la complexité théorique reste exponentielle, ces optimisations rendent l'exploration plus tractable dans des cas pratiques.

2.4 Introduction des Heuristiques

L'exploration complète d'un arbre de jeu souffre souvent de l'explosion combinatoire. Une optimisation consiste à transformer l'arbre en un graphe en évitant de ré-explore des sous-graphes communs, ce qui nécessite de mémoriser les nœuds déjà explorés. Cependant, cette approche peut devenir coûteuse en temps et en mémoire, en particulier pour des jeux complexes. Une autre amélioration consiste à introduire des **coupes intelligentes** : si une branche garantit une victoire (Joueur 1) ou une défaite (Joueur 2), les branches voisines sont ignorées. Cela réduit la taille de l'arbre tout en conservant les propriétés de l'exploration complète.

Pour aller plus loin, on limite l'exploration à une **profondeur maximale** p , où les feuilles de l'arbre sont évaluées à l'aide d'une **heuristique**. Une heuristique est une fonction statique qui attribue une valeur numérique à un plateau de jeu, reflétant sa qualité pour un joueur. Elle est positive lorsque l'état se rapproche de la victoire, négative lorsqu'il se rapproche de la défaite, et cohérente avec les scores réels dans les configurations terminales. En pratique, l'heuristique permet de guider l'exploration en priorisant les branches prometteuses, réduisant ainsi la complexité sans explorer tout l'arbre.