



UFR MATHÉMATIQUES ET INFORMATIQUE

Rapport de projet d'intelligence artificielle

Hasami Shōgi

Réalisé par :

Ben Slama Sana et Omar Amaraa

Encadrante :

Elise Bonzon

Année universitaire 2024–2025

Table des matières

1	Introduction	2
2	Description du jeu	2
2.1	Description générale	2
2.2	Objectif du jeu	2
2.3	Déplacement des pions	2
2.4	Capture des pions	3
2.5	Déroulement de la partie	3
3	IA implémentées	4
3.1	Niveaux d'IA	4
3.2	IA Niveau 1	4
3.3	IA Niveau 2	4
3.4	IA Niveau 3	4
3.5	IA Niveau 4	5
3.6	Fonctions d'évaluation	5
3.7	Optimisations techniques	5
4	Tournois entre IAs	6
4.1	Performance globale par IA	6
4.2	Tête-à-tête : taux de victoires du premier joueur (Noir)	6
4.3	Taux de victoire selon le rôle (Noir vs Blanc)	7
5	Bilan	9
6	Sources et bibliographie	10

1 INTRODUCTION

Dans le cadre de notre projet, nous avons choisi d'explorer un jeu déterministe à information parfaite : le **Hasami Shogi**. Cette variante populaire du Shogi, le jeu d'échecs japonais, est particulièrement appréciée pour sa simplicité et ses mécanismes stratégiques uniques. Originaire du Japon, son nom, « hasami », qui signifie « pince », fait directement référence à la méthode de capture des pions adverses par une prise en tenaille.

Notre objectif est de concevoir une intelligence artificielle avec plusieurs niveaux de difficulté capable de jouer efficacement à ce jeu.

2 DESCRIPTION DU JEU

2.1 Description générale

Le Hasami Shogi se joue sur un plateau de 9x9 cases, semblable au Shogi classique. Chaque joueur commence avec 9 pions, qui sont alignés sur la première rangée devant lui.

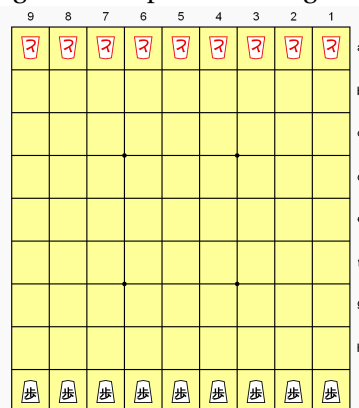


FIGURE 1 – Aperçu du plateau de Hasami Shogi (Attention sur la figure deux lignes sont affiché)

2.2 Objectif du jeu

Le Hasami Shogi est un jeu de stratégie où chaque joueur vise à capturer les pions adverses en utilisant la prise en tenaille, tout en protégeant les siens.

La partie se termine lorsqu'un joueur n'a plus que deux pions ou moins, ou si l'écart entre les deux joueurs dépasse trois pions.

2.3 Déplacement des pions

Voici les règles de déplacement des pions :

- Les pions se déplacent comme une tour aux échecs, soit horizontalement ou verticalement sur un nombre illimité de cases.
- Il est interdit de sauter par-dessus un pion, qu'il soit allié ou ennemi.
- Chaque tour, le joueur est obligé de déplacer un pion, sauf si aucun déplacement n'est possible. Dans ce cas, le joueur passe son tour.

2.4 Capture des pions

Dans le Hasami Shogi, la capture des pions repose sur un principe de prise en tenaille, qui peut s'appliquer à un ou plusieurs pions selon leur position sur le plateau.

Dans notre version du jeu, nous avons introduit des options supplémentaires qui ne figurent pas dans les règles traditionnelles du jeu. Ces options peuvent être activées ou désactivées selon les préférences.

1. **La capture d'un pion** : Un pion ennemi est capturé s'il est pris en tenaille, c'est-à-dire entouré par deux pions alliés horizontalement ou verticalement.
2. **La capture d'une suite de pions** : Une série de pions consécutifs peut être capturée si leurs deux extrémités sont occupées par des pions alliés.
3. **La capture dans les coins du plateau** (optionnelle) : Un pion placé dans un coin peut être capturé si ses deux cases adjacentes sont occupées par des pions adverses.
4. **La capture diagonale** (optionnelle) : Un pion ennemi peut être capturé si deux pions alliés l'encadrent sur une même diagonale.

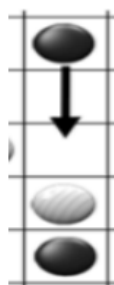


FIGURE 2 – Illustration d'une capture verticale

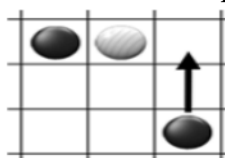


FIGURE 3 – Illustration d'une capture horizontale

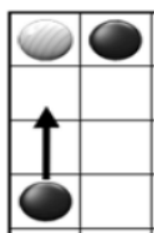


FIGURE 4 – Illustration d'une capture dans le coin

2.5 Déroulement de la partie

Les joueurs jouent à tour de rôle, chaque joueur devant obligatoirement déplacer un de ses pions à son tour. Si un joueur se retrouve dans l'incapacité de déplacer un pion, il passe son tour.

Les captures se font par prise en tenaille : un pion ou une série de pions adverses est capturée lorsqu'ils sont entourés par deux pions alliés, que ce soit horizontalement ou verticalement, en fonction des règles activées dans la partie (comme la capture diagonale ou dans les coins, si ces options sont activées).

La partie prend fin lorsqu'un joueur perd tous ses pions sauf deux, ou moins. Dans ce cas, l'adversaire est déclaré vainqueur. Il est également possible de gagner si la différence entre le nombre de pions des deux joueurs dépasse trois, ce qui entraîne une défaite automatique pour celui ayant le moins de pions.

Le vainqueur est généralement celui qui parvient à combiner une stratégie défensive efficace et une attaque réfléchie, capturant les pions adverses tout en minimisant ses propres pertes.

3 IA IMPLÉMENTÉES

Dans le cadre de notre projet Hasami Shogi, quatre niveaux d'intelligence artificielle ont été développés, tous basés sur l'algorithme Minimax, avec ou sans élagage *Alpha-Beta*, et dotés de fonctions d'évaluation et de pruneurs adaptés.

3.1 Niveaux d'IA

3.2 IA Niveau 1

- **Profondeur fixe** : 3 plies (réduite à 2 en phase d'optimisation).
- **Recherche** : Minimax classique sans α - β .
- **Évaluation** : `evaluer_position_naive` — simple différence de matériel, petit bonus pour le contrôle du centre, et comptage basique des menaces horizontales/verticales.
- **Move ordering** : tri léger pour prioriser les captures, sans mémorisation.
- **Usage** : IA très rapide, adaptée aux tests ou aux joueurs débutants, mais limitée tactiquement.

3.3 IA Niveau 2

- **Profondeur fixe** : 4 plies.
- **Recherche** : Minimax avec élagage α - β .
- **Évaluation** : `evaluer_position_naive`.
- **Améliorations** : mémorisation via `functools.lru_cache` pour éviter de recalculer des positions déjà explorées.
- **Usage** : IA plus rapide et plus efficace que le niveau 1, mais toujours limitée par une évaluation simplifiée.

3.4 IA Niveau 3

- **Profondeur fixe** : 4 plies.
- **Recherche** : élagage α - β .
- **Évaluation** : `evaluer_position`, incluant :
 - Matériel et contrôle du centre,
 - Mobilité (nombre de coups disponibles),
 - Menaces avancées (captures imminentes),

-
- Groupes de pions adjacents,
 - Contrôle des coins.
 - **Optimisations** : mémoïsation LRU, move ordering (captures et centre en priorité).
 - **Usage** : IA stratégique et équilibrée, offrant un bon compromis entre temps de réponse et profondeur.

3.5 IA Niveau 4

- **Profondeur dynamique** : 3 plies en début de partie, jusqu'à 5 en fin de partie selon la phase de jeu.
- **Recherche** : α - β avec *futility pruning* (si l'évaluation est très mauvaise, on choisit un coup "greedy"), et priorité aux coups centraux en ouverture.
- **Évaluation** : même `evaluer_position` que le niveau 3.
- **Optimisations avancées** :
 - Mémoïsation LRU sur les positions explorées,
 - Move ordering amélioré (captures avant centre),
 - Heuristique d'ouverture (choix aléatoire parmi les coups centraux),
 - Pruneurs statiques (*futility pruning* et seuils sur les scores),
- **Usage** : moteur le plus complet, combinant solidité, adaptabilité et performances.

3.6 Fonctions d'évaluation

- `evaluer_position_naive` (niveaux 1 et 2) : différence de pions, contrôle léger du centre, menaces simples.
- `evaluer_position` (niveaux 3 et 4) : reprend l'approche naïve puis ajoute mobilité, menaces avancées, structure des groupes et contrôle des coins.

3.7 Optimisations techniques

Afin d'améliorer la rapidité d'exécution, la pertinence des coups joués et la diversité des parties, plusieurs optimisations ont été mises en place dans notre moteur d'IA :

Mémoïsation des évaluations (LRU cache) : toutes les évaluations de positions sont mises en cache à l'aide du décorateur `functools.lru_cache`. Cela permet d'éviter le recalcul de positions déjà explorées au cours de l'arbre de recherche, ce qui améliore considérablement les performances, notamment lorsqu'une même configuration est atteinte via plusieurs séquences différentes.

Ordonnancement des coups (Move ordering) : les coups sont triés de manière à traiter en priorité les captures et les déplacements vers les cases centrales du plateau. Cette stratégie améliore l'efficacité de l'élagage α - β , en maximisant la probabilité de couper des branches dès les premières explorations. Cela permet aussi à l'IA de repérer plus rapidement les coups stratégiques.

Ouvertures aléatoires centrées : lors des tout premiers coups (notamment dans les 9 premiers déplacements), l'IA sélectionne aléatoirement un coup parmi ceux situés dans la zone centrale du plateau. Cela permet d'éviter que toutes les parties démarrent de la même façon, et introduit une diversité intéressante sans compromettre la qualité stratégique de l'ouverture.

Pruneurs statiques (futility pruning) : lorsque l'évaluation statique d'une position est déjà très défavorable (par exemple inférieure à -5), certaines branches sont abandonnées immédiatement (futility pruning). Cela permet de ne pas explorer inutilement des variantes qui mèneraient de toute façon à une position perdante, économisant ainsi du temps de calcul. Exemple :

- On est en train d'explorer un coup à une profondeur limitée (ex. profondeur 1 ou 2),
- On évalue la position : elle est très mauvaise (ex. notre score est déjà < -5),
- Alors on n'explores même pas les suites de coups à partir de cette position.

Élagage sur chute de score : lorsqu'un coup amène à une position clairement inférieure par rapport à l'évaluation actuelle, l'IA peut l'ignorer si la perte est jugée trop importante. Cette logique repose sur l'idée qu'un moteur fort n'a pas intérêt à transiter par une position objectivement inférieure s'il possède un avantage confortable.

Utilisation contrôlée du hasard : dans les cas où plusieurs coups obtiennent une évaluation identique, un choix aléatoire est effectué parmi eux. Cela permet de ne pas toujours sélectionner le même coup par défaut et de varier les parties, tout en restant dans le cadre de coups jugés équivalents en termes de qualité.

Profondeur adaptative : la profondeur de recherche n'est pas fixe pour l'IA la plus avancée. Elle s'adapte à la phase de jeu (début, milieu ou fin), au nombre de pions restants et au niveau de l'adversaire. Cela permet de jouer plus rapidement en ouverture ou contre des IA faibles, tout en augmentant la profondeur dans les situations critiques.

4 TOURNOIS ENTRE IAS

4.1 Performance globale par IA

Observations :

- **IA 1** : $\approx 10\%$ de victoires, 70% de défaites, 20% de nuls.
- **IA 2** : $\approx 30\%$ de victoires, 50% de défaites, 20% de nuls.
- **IA 3** : $\approx 55\%$ de victoires, 25% de défaites, 20% de nuls.
- **IA 4** : $\approx 75\%$ de victoires, 15% de défaites, 10% de nuls.

Justifications :

1. La profondeur de recherche et les optimisations (alpha-beta, tables de transposition, pruneurs) augmentent avec le niveau, permettant aux IA supérieures de transformer systématiquement de petits avantages en victoires.
2. Le taux de nuls (*draw rules*) reste autour de $15\text{--}20\%$, résultat des positions où aucune IA ne trouve de coup gagnant dans la profondeur allouée.
3. L'initiative (joueur premier) confère un léger avantage, visible dans les résultats globaux.

4.2 Tête-à-tête : taux de victoires du premier joueur (Noir)

Observations :

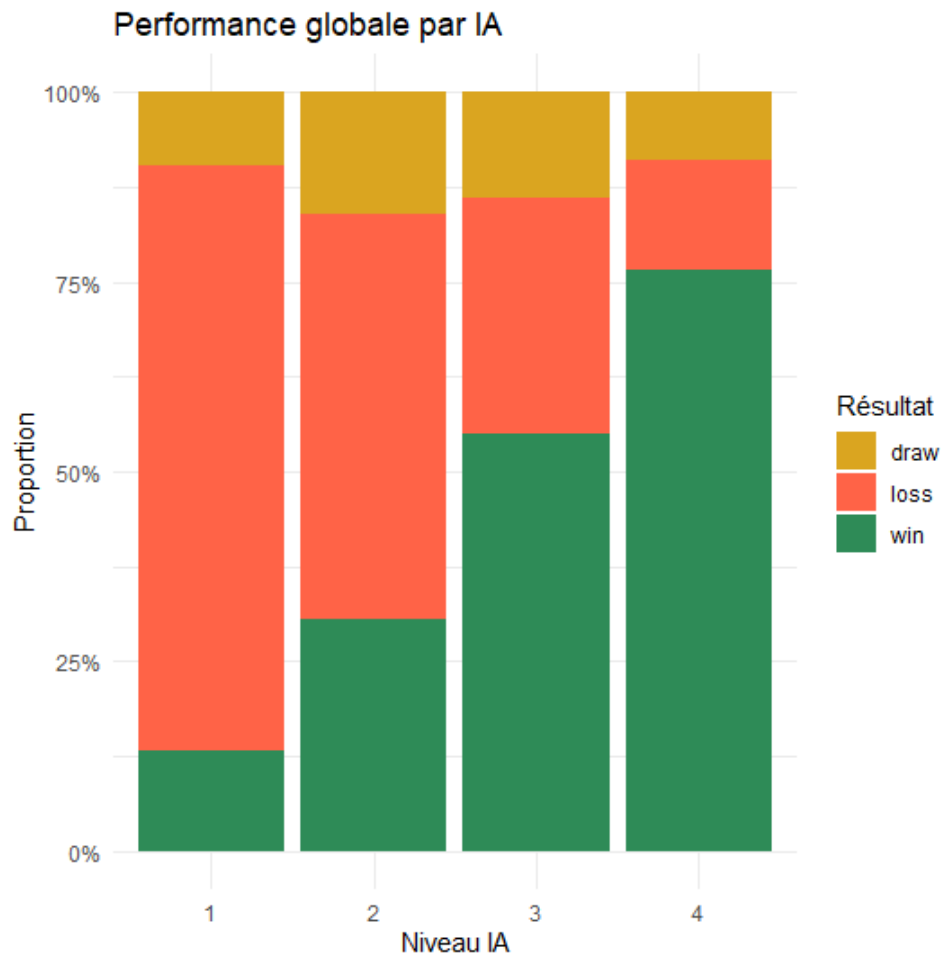


FIGURE 5 – Répartition des victoires, défaites et nuls pour chaque niveau d'IA, tous rôles confondus.

- Pour un écart de niveau important (e.g. Noir 4 vs Blanc 1), le premier joueur gagne dans plus de 85% des parties.
- Pour des niveaux voisins (e.g. Noir 2 vs Blanc 3), le taux oscille autour de 50–60%, signe d'un affrontement équilibré.
- L'avantage du premier coup s'atténue dès que la profondeur de recherche de l'adversaire devient proche.

Justifications :

- Plus l'écart de profondeur est grand, plus l'initiative initiale se convertit aisément en avantage matériel et en victoire.
- À niveaux proches, l'avantage du premier coup n'est plus suffisant pour dépasser la limite de recherche adverse.

4.3 Taux de victoire selon le rôle (Noir vs Blanc)

Observations :

- Chaque IA gagne légèrement plus souvent en tant que joueur Premier (Noir) qu'en tant que Second (Blanc), l'écart variant de 1 à 2 points de pourcentage.

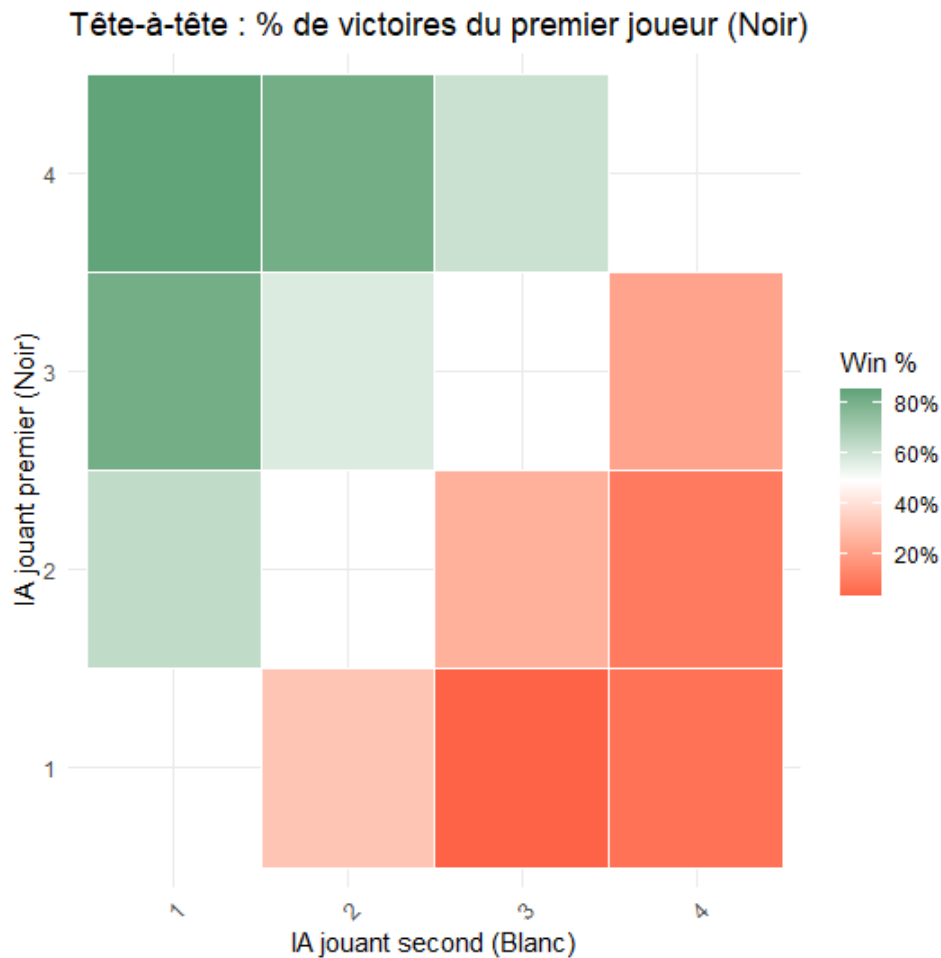


FIGURE 6 – Pourcentage de victoires du joueur premier (Noir) selon les niveaux face à l’adversaire second (Blanc).

Justifications :

- Le coup initial offre un micro-avantage positionnel (initiative) impactant légèrement le taux de victoire.
- Le tie-break aléatoire et la profondeur de recherche limitent cet avantage à quelques points.

CONCLUSION

Les tournois valident que :

- L’initiative (premier joueur) procure un léger avantage.
- La profondeur de recherche et la fonction d’évaluation restent les principaux leviers pour accroître significativement le taux de victoires. Cependant, un IA beaucoup moins forte parvient quand même parfois à battre une IA beaucoup plus forte (en théorie).
- Les règles de nuls assurent un nombre non négligeable de matchs nuls lorsque les forces sont comparables. D’ailleurs c’est souvent le cas.

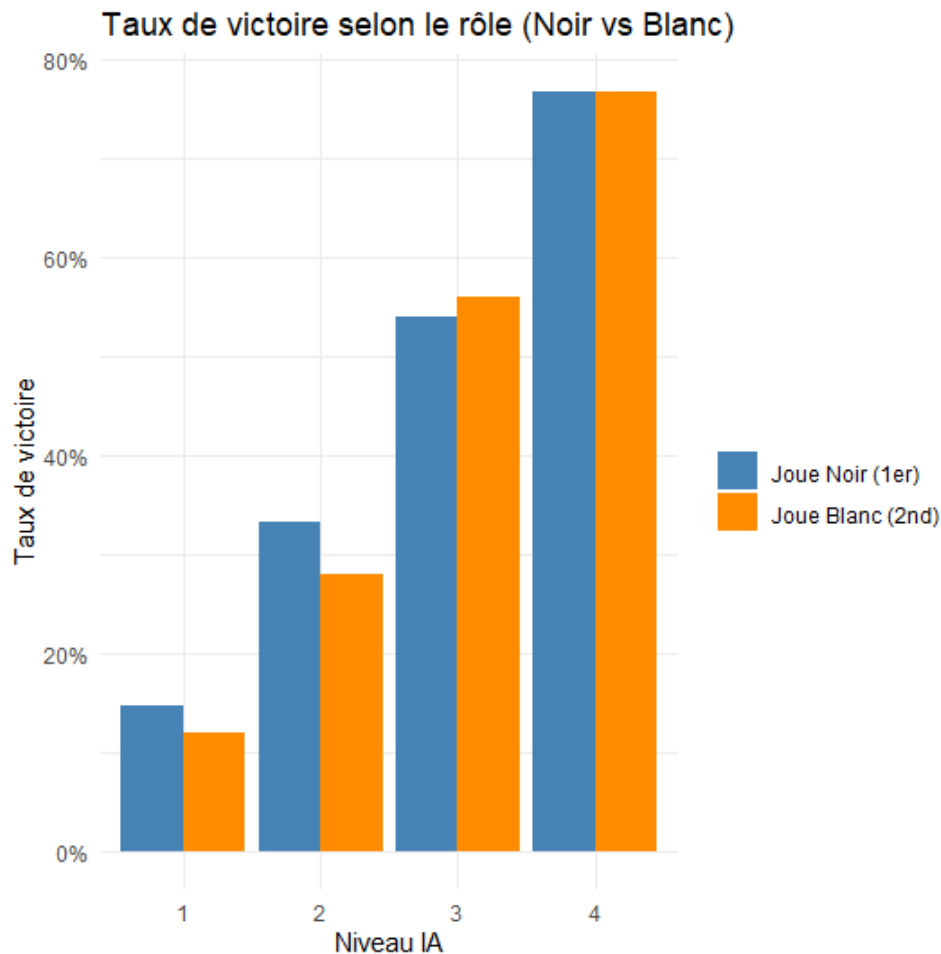


FIGURE 7 – Comparaison des taux de victoire d’une même IA selon qu’elle joue Premier (Noir) ou Second (Blanc).

5 BILAN

Malgré les différences significatives de profondeur de recherche, d’optimisation et de sophistication algorithmique entre les différentes IA testées, certains résultats inattendus ont été observés.

En particulier, une IA beaucoup moins puissante — reposant sur un MiniMax à faible profondeur et une évaluation naïve — parvient parfois à surprendre une IA de niveau 4 équipée d’alpha-bêta, de tables de transposition et de pruneurs avancés. Cela s’explique notamment par l’absence de support théorique : contrairement aux échecs, où des bases d’ouvertures et des bases de finales (« tablebases ») permettent d’éviter l’exploration de positions nulles ou sans issue, notre moteur de Hasami-Shogi doit redécouvrir chaque configuration à partir de zéro, ce qui le rend plus vulnérable à certaines séquences atypiques ou à des pièges simples mais efficaces.

Ce constat souligne l’intérêt d’intégrer, à terme, une gestion plus théorique des ouvertures ou des motifs positionnels caractéristiques afin de renforcer la stabilité décisionnelle de l’IA.

Malgré cela, les résultats globaux montrent que l’IA de niveau 4 conserve un avantage net et constant sur l’ensemble des autres niveaux, confirmant l’efficacité de l’ensemble des optimisations mises en place dans son moteur.

6 SOURCES ET BIBLIOGRAPHIE

- **Minimax algorithm** — *Wikipedia*. Disponible en ligne :
<https://en.wikipedia.org/wiki/Minimax>
- **Alpha-beta pruning** — *Wikipedia*. Disponible en ligne :
https://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning
- **Futility pruning** — *Wikipedia*. Disponible en ligne :
https://en.wikipedia.org/wiki/Futility_pruning
- **Futility pruning (technique avancée)** — *ChessProgramming Wiki*. Disponible en ligne :
https://www.chessprogramming.org/Futility_Pruning
- **Hasami Shogi** — *Wikipedia*. Disponible en ligne :
https://en.wikipedia.org/wiki/Hasami_Shogi
- **Règles du Hasami Shogi** — *BrainKing*. Disponible en ligne :
<https://brainking.com/en/GameRules?tp=73>
- **Tutoriel vidéo : How to Play Hasami Shogi** — *YouTube*. Disponible en ligne :
<https://www.youtube.com/watch?v=hI5G6Yjcgjc>
- **AlphaGo — The Movie (Full Documentary)** — *DeepMind / YouTube*. Disponible en ligne :
<https://www.youtube.com/watch?v=WXuK6gekU1Y>
- **Introduction to Game-Playing AI (Minimax, Alpha-Beta)** — *MIT OpenCourseWare*. Disponible en ligne :
<https://www.youtube.com/watch?v=STjW3eH0Cik&t=447s>
- **How a Chess Engine Evaluates Positions** — *Chessify Blog*. Disponible en ligne :
<https://chessify.me/blog/chess-engine-evaluation>
- **ELI5 : How does a chess engine determine how good a position is?** — *Reddit*. Disponible en ligne :
https://www.reddit.com/r/explainlikeimfive/comments/18nb0j7/eli5_how_does_a_chess_engine_determine_how_good/?rdt=58366