

Stratégies optimales et approches computationnelles dans le jeu « Bataille navale »

Revue de littérature

Introduction

Le jeu de la *Bataille navale* (ou *Battleship* en anglais) est un jeu de société combinatoire à deux joueurs, où chacun dispose une flotte de navires sur une grille généralement de 10 *times* 10 cases, puis tente de deviner les positions des navires adverses en appelant des tirs sur des coordonnées. C'est un jeu à information **imparfaite** : les positions des navires de l'adversaire sont cachées tout au long de la partie, ce qui place ce jeu dans la catégorie des jeux « à brouillard de guerre » similaires à la version *Kriegspiel* des échecs ou à Stratego. Chaque tir révèle partiellement l'état du jeu (« à l'eau » pour un échec, « touché » pour un impact, et éventuellement « coulé » lorsqu'un navire est entièrement détruit), ce qui guide progressivement les décisions du joueur.

Malgré des règles simples, la Bataille navale présente des défis stratégiques et algorithmiques profonds. D'une part, elle constitue un problème de recherche séquentielle dans un espace d'états partiellement observable, ce qui en fait un cas d'étude intéressant pour l'intelligence artificielle (IA) : un agent artificiel doit y planifier ses actions de tir en tenant compte de l'incertitude sur l'état réel (placements adverses) et mettre à jour ses croyances (*belief state*) après chaque observation. D'autre part, en théorie des jeux, la Bataille navale est un jeu à somme nulle où chaque joueur pourrait adopter des stratégies mixtes (notamment pour le placement des navires) pour maximiser ses chances de victoire. L'étude du jeu soulève donc des questions sur l'**optimisation des stratégies d'attaque** (recherche des navires adverses en un minimum de coups) et des **stratégies défensives** (placement optimal des navires pour échapper aux tirs adverses). Cette revue de littérature synthétise les contributions majeures sur ces aspects, en couvrant (i) les résultats théoriques sur la complexité et les stratégies optimales, (ii) les heuristiques pratiques et modèles probabilistes développés, (iii) les approches par apprentissage et recherche Monte Carlo en IA, et (iv) les principes d'un bon placement défensif.

Contributions théoriques et modèles formels

Du point de vue théorique, déterminer une stratégie optimale de tir qui minimise le nombre de coups nécessaires pour trouver tous les navires adverses est un problème difficile. Plus formellement, on peut modéliser la recherche d'un navire (ou d'une flotte de navires) caché(s) comme un problème de construction d'un arbre de décision binaire : chaque nœud représente un tir (avec deux issues possibles : succès ou échec) et les feuilles

correspondent à des configurations complètement localisées. Trouver l'arbre de décision de profondeur minimale qui distingue toutes les configurations possibles de navires est un problème NP-complet en général. En effet, Hyafil et Rivest ont démontré dès 1976 que la construction d'arbres de décision optimaux est NP-complete, ce qui s'applique en particulier à la « stratégie 20 questions » de la Bataille navale. Par conséquent, une stratégie déterministe garantissant un nombre minimal de coups au pire cas n'est pas calculable efficacement pour des grilles de grande taille, sauf à explorer un espace exponentiel de possibilités.

Heuristiques de recherche et modèles probabilistes

Face aux limitations des approches purement déterministes, de nombreuses **heuristiques** ont été proposées pour améliorer l'efficacité moyenne des recherches de navires. Historiquement, les joueurs humains ont développé empiriquement des méthodes telles que :

- **Le tir aléatoire contrôlé + ciblage** (*Hunt and Target*) : Cette stratégie, sans doute la plus instinctive, consiste à tirer de façon aléatoire (ou en quadrillant grossièrement la grille) tant qu'aucun navire n'est repéré (« phase de chasse »), puis, lorsqu'un tir est « touché », à tirer autour de cette case pour trouver l'orientation et couler le navire touché (« phase de ciblage »). Une fois le navire coulé, on retourne en mode chasse. C'est la stratégie de base enseignée aux nouveaux joueurs. Berry a mesuré qu'elle demande en moyenne 66 coups pour gagner contre un placement aléatoire.
- **Le balayage par motifs (parité)** : Comme mentionné plus haut, tirer seulement sur les cases alternées (couleur d'échiquier) permet de ne rater aucun navire de longueur *ge2*. On peut également employer d'autres motifs de balayage (par exemple, des bandes diagonales, ou un espacement adaptatif en fonction de la plus petite taille de navire restante). Ces méthodes visent à optimiser la couverture de l'espace des possibles. La différence d'efficacité est modeste mais réelle : le damier réduit légèrement le nombre moyen de coups par rapport à un balayage aléatoire simple (65 au lieu de 66 dans l'analyse de Berry).
- **Le ciblage systématique** : Une fois un navire touché, une heuristique efficace consiste à cibler en priorité les cases adjacentes alignées, pour rapidement couler le navire. Si l'on touche une case mais que les cases voisines directes sont des « touches » négatives (à l'eau), cela indique que le navire touché est de taille 2 et est coulé, ou qu'on a atteint une extrémité d'un navire plus grand. Cette stratégie de ciblage fait partie intégrante des algorithmes heuristiques standard et réduit drastiquement les coups gaspillés après une première touche.

Approches en intelligence artificielle

La Bataille navale a récemment servi de terrain d'expérimentation pour des méthodes d'intelligence artificielle, notamment en **apprentissage automatique**. Deux approches principales ont été explorées : l'apprentissage supervisé de stratégies à partir de données, et l'apprentissage par renforcement pour qu'un agent apprenne à jouer par lui-même en optimisant son taux de victoire. En outre, des techniques de recherche générale comme la **recherche arborescente Monte Carlo** (MCTS) ont été appliquées pour gérer l'incertitude du jeu.

Stratégies défensives et optimisation du placement

Jusqu'ici, nous avons surtout envisagé un joueur attaquant face à un placement adverse *aléatoire*. Or, du point de vue du joueur en défense (celui qui place ses navires), il est légitime de chercher une configuration qui *maximise* la difficulté de l'adversaire. Autrement dit, existe-t-il un placement optimal des navires qui prolonge la durée moyenne de la partie en cas d'attaque adverse optimale ? Cette question peut être abordée par une analyse statistique et algorithmique.

Conclusion

La Bataille navale, sous des dehors ludiques, se révèle être un problème riche en enseignements pour l'intelligence artificielle et la théorie des jeux. Cette revue a mis en lumière plusieurs apports majeurs. D'abord, sur le plan théorique, la recherche exhaustive de stratégies optimales s'avère complexe (NP-complet pour la formulation en arbre de décision), ce qui justifie l'usage d'heuristiques. Ensuite, des **stratégies heuristiques et probabilistes** efficaces ont été développées : l'approche par fonction de densité de probabilité de Berry a établi un standard en réduisant drastiquement le nombre moyen de coups nécessaires (autour de 44). Cette stratégie exploite pleinement les informations partielles obtenues, montrant qu'un raisonnement bayésien peut surpasser les techniques humaines standard.

Références

- [1] M. Audinot, F. Bonnet, and S. Viennot. Optimal strategies against a random opponent in battleship. In *19th Game Programming Workshop*, pages 67–74, 2014.
- [2] Nick Berry. Battleship strategy : mathematical analysis of the classic game. Blog DataGenetics, 2011. Disponible sur <http://datagenetics.com/blog/december32011/index.html>.

- [3] C. L. Brown. Battleship probability calculator : Methodology. [Online] Disponible sur <https://cliambrown.com/battleship/methodology.php>, 2015.
- [4] GA-Intelligence. Deep reinforcement learning – how to win at battleship. Blog en ligne, 2019. <https://www.ga-intelligence.com/deep-reinforcement-learning-win-battleship>.
- [5] S. Gelly and Y. Wang. Introduction to monte carlo tree search. In *Synthese*, pages 57–69. 2012.
- [6] C. Houghton. Battle-ship-ai. Project GitHub, 2020. <https://choughton21.github.io/Battle-Ship-AI/>.
- [7] Laurent Hyafil and Ronald L. Rivest. Constructing optimal binary decision trees is np-complete. *Information Processing Letters*, 5(1) :15–17, 1976.
- [8] Chanbin Park. Optimal strategies on attack and defense in the game of battleship. Acton Scholars research report 06-Optimal_Strategies_Battleship-200902, 2020.
- [9] Stuart Russell and Peter Norvig. *Intelligence Artificielle – Une approche moderne*. Pearson Education France, 3 edition, 2010. Section 5.6.1, Jeux partiellement observables.
- [10] J. Sansom et al. Using supervised training to teach a network to play battleship. Medium article, 2021. <https://medium.com/@jayceesansom/using-supervised-training-to-teach-a-network-to-play-battleship-7522f28c52ca>.