Deciphering Enemies in the Darkness through Modeling and Examination of Knowledge in Reconnaissance Blind Chess

Robin Stöhr [1] **Shuai Wang** [1]

Abstract

An important research topic about Theory of Mind (ToM) is the ability to understand and reason about how agents acquire and predict the behavioral and mental states of other agents in dynamic environments, especially those involving a significant change in knowledge and information. In this paper, we focus on the modeling and examination of knowledge of other agents in imperfect information games. More specifically, we delve into the nuances of the change of knowledge in the Reconnaissance Blind Chess (RBC). In each round, players are granted limited sensing capacity of the board. Thus, the understanding opponent's knowledge and strategy plays a key role in decision-making in each round. This paper studies how an agent can model and utilize an opponent's knowledge in the RBC game. The examination includes a detailed comparison of information obtained through different actions in the game. We design two sensing strategies for obtaining information based on entropy and other factors and compare how these strategies can impact the outcome of the game. Finally, we discuss how our research results could be generalized to the understanding of opponents' knowledge and behavior in non-cooperative imperfect information games.

1. Introduction

The capability of understanding and reasoning about how agents acquire and predict the behavioral and mental states of other agents remains a major challenge in A.I. More uncertainty arises in a dynamic environment involving a significant change in knowledge and information. Theory of Mind (ToM) refers to the ability to understand and reason about the mental states, decisions, and emotions of other agents. It has been studied from various perspectives, such as cognitive science (Carlson et al., 2013) and social science (Carruthers & Smith, 1996). ToM plays a crucial role in the field of computer science. For instance, Aru et al. highlight the importance of ToM in the development of deep learning systems (Aru et al., 2023). The authors argue that understanding ToM is essential for creating artificial agents capable of effectively interacting with humans as well as providing unique insights into the complexity of ToM that are challenging to study in humans.

The information we can obtain to understand an agent's mental state can vary greatly depending on the context. In some cases, we can gather this information through communication, observation of actions and their outcomes, and so on. Some games can be adapted for the evaluation of models for the understanding of the decisions of both humans and agents in games. Essentially, these games can serve as platforms for designing and testing strategies for modeling knowledge. Moreover, these games make it easy to evaluate the performance of complex strategies and agents designed in different approaches (e.g. Logic-based approach v.s. neural-network-based approach). In this paper, we focus on Reconnaissance Blind Chess (RBC). a variant of chess with imperfect information. In each round, players are granted limited sensing capacity of the board. This introduces uncertainty in terms of knowledge about the opponent's pieces. RBC provides an ideal platform for exploring and evaluating frameworks inspired by ToM. Understanding an opponent's knowledge is critical in RBC. A deep comprehension of what an opponent knows can inform not only accurate predictions of their future moves but also intelligent strategies that can capture the opponent's pieces.

Despite that the RBC game has been studied, little work was done for the measuring of the gain of information through various actions and how this impacts the strategies. This paper presents preliminary work on a comprehensive examination and modeling of an opponent's knowledge in RBC. More specifically, we study the following research questions:

[1]Department of Computer Science, Vrije Universiteit Amsterdam, Amsterdam, the Netherlands. Correspondence to: Robin Stöhr <r.stohr@student.vu.nl>, Shuai Wang <shuai.wang@vu.nl>.

Proceedings of the First Workshop on Theory of Mind in Communicating Agents, Honolulu, Hawaii, USA. PMLR 202, 2023.

RQ1: How much information about the opponent does the agent obtain through different actions in the game?

RQ2: How do different sensing strategies impact the knowledge of the opponent and the result of the game?

The paper is structured as follows: Section 2 introduces the rules and winning conditions of RBC. Section 3 discusses our detailed modeling and analysis of methods for obtaining and updating game state knowledge. Section 4 explains how we use our knowledge to compute sensing strategies. Section 5 details the moving strategy, board evaluation, and other implementation aspects. Section 6 presents the evaluation results. These research questions are important in general for non-cooperative games with imperfect information. Finally, we provide some discussion and future work in Section 7.

2. Reconnaissance Blind Chess

Reconnaissance Blind Chess (RBC) is a variant of traditional chess that adds an additional layer of complexity by introducing uncertainty about the opponent's knowledge of board configuration. Unlike classic chess, where all pieces are visible to both players, in RBC, players are blind to their opponent's pieces and must perform a "sensing" step each round to learn about a 3x3 region of the board. This information is private, with the opponent remaining unaware of the player's sensing location.

When capturing a piece, players are only notified of the location of the capture, not the type of piece that was captured. This differs from classical chess where both the type and position of the captured piece are revealed.

In the case of illegal moves, such as moving a pawn diagonally to an empty square, the move is deemed unsuccessful and the player's turn ends. If a player successfully moves a sliding piece through an opponent's piece, the opponent's piece is captured and the moved piece stops where the capture occurred.

The game also includes time constraints similar to traditional chess. Each player begins with a 15-minute clock and gains 5 seconds after each turn. If 50 turns pass without a pawn move or capture, the game is declared a draw. Players also have the option to "pass" a turn. The game is won either by capturing the opponent's king or if the opponent runs out of time.

To provide a better understanding of RBC, we present an example game that was played between StrangeFish2 (white) and Trout (black)[1]. This game showcases key aspects of RBC, including sensing, piece movement, and capturing the opponent's king. Please refer to Figure **??** for illustrations of these concepts.

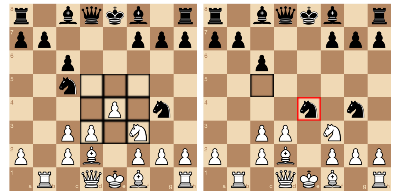

Figure 1. Example of capturing. Black senses the 3x3 square on e4, and subsequently moves c5→e4

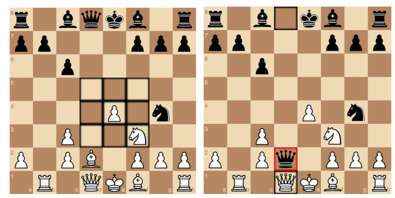

Figure 2. Example of an interrupted move. Black senses, and then tries to move the queen d8→d1. But as there is a bishop on d2, the move stops there and the bishop is captured.

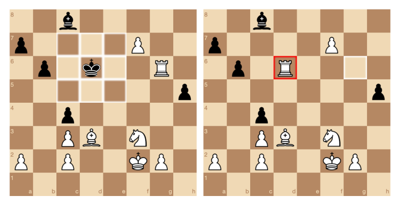

Figure 3. Example of an ending turn. White senses the location of the enemy king and captures the king with the rook.

3. Knowledge Modelling

The main difference between RBC and classical chess lies in the uncertainty of information within the game. A key knowledge in RBC is the information that a player has about its opponent. At each round, there are three distinct moments when a player receives new information about the game state: the sensing result, move result, and opponent move result.

Following the receipt of these notifications, a player updates its knowledge according to the new information. Specifically, we can track the knowledge of a player exhaustively by listing all possible game states and expanding or reducing this list according to the new information. At each time, in the game, the player maintains a set of possible states as its knowledge. For a transition to the next round, we expand each state and keep only those expanded states that align with the opponent's move results. The complete flow of the notifications and corresponding actions can be seen in Figure 4.

This comprehensive model enables a player to optimize its actions based on a detailed understanding of the opponent's potential knowledge and strategies, exemplifying the

[1]The replay of the game can be seen at https://rbc.jhuapl.edu/games/628294

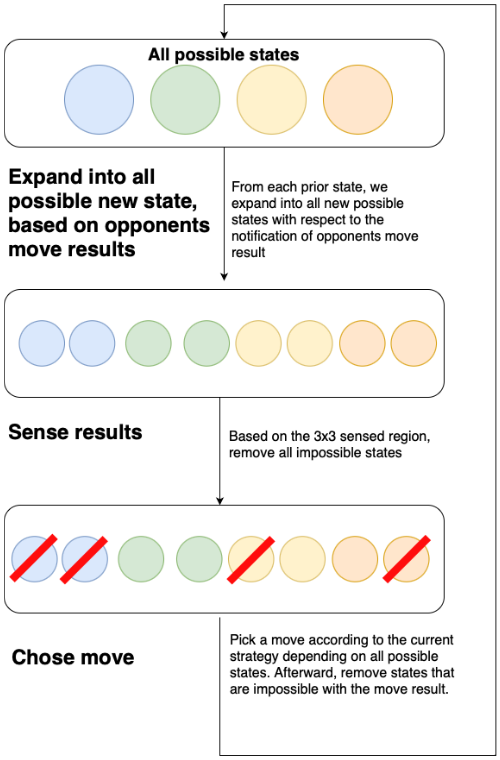

Figure 4. Management of possible states and their use for the decisions of sensing and moving

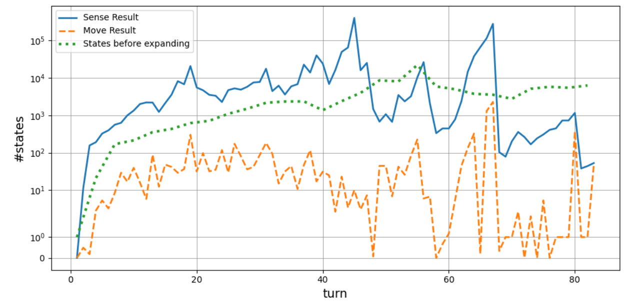

Figure 5. Number of removed states after sensing, moving compared against the number of remaining states. Omitted in this plot are the states added by expanding, after the opponents turn.

application of Theory of Mind in game scenarios.

3.1. Knowledge modeling for sensing and moving

The sensing phase of the game is crucial for gaining knowledge about the opponent's game state. When an agent selects a 3x3 region to sense, it is provided with information about the true pieces in that region. This effectively narrows down the possible game states that align with the agent's knowledge, as any states inconsistent with the sensed information are removed from consideration (Figure 1).

The agent's knowledge is also updated after a move is executed. The result of a move, including whether it was successful, whether an enemy piece was captured, and the square on which the capture occurred, is relayed to the agent. However, the type of the captured piece is not revealed. If a capture happens at a location different from the intended destination, the agent's piece stays at the capture square (Figure 2). This new information allows further refinement of the agent's knowledge, as game states that conflict with the move results are excluded. For example, if a piece was captured, all game states that do not contain a piece on the capture square are discarded. Similarly, if a sliding move was successful, all squares with a piece between the original and destination squares are purged from consideration.

In certain instances, there can be overlapping information obtained from the sensing and moving phases, such as when the 3x3 region sensed overlaps with the trajectory of the agent's move. To study the first research question (RQ1), we evaluated the knowledge obtained from different actions and observations by measuring the number of moved states. We compared the gain in information by examining the number of states removed due to sensing and moving. As Figure 5 indicates, sensing can remove a significantly greater number of states than moving[2]. This is because sensing provides complete knowledge of nine squares (including the type and location of pieces), while moving can provide at most partial information about seven squares (either empty squares or an unknown piece type). Thus, a key aspect of gaining information in RBC hinges on a well-strategized sensing approach.

3.2. Knowledge from opponent's move

The information provided after an opponent's move is limited to whether one of the agent's own pieces was captured and, if so, the square on which this occurred. Armed with this information, we consider all new possible states, based on all the moves the opponent could have executed in all possible positions. However, in this paper, we do not adjust our knowledge based on our best guess of the opponent's moving strategy, owing to its complexity.

4. Sensing Strategies

To address the second research question (RQ2), we explore how different sensing strategies can impact the knowledge of the opponent and the result of the game. In Reconnaissance

[2]To standardize the metrics, we utilized the naive entropy sensing mechanism within the game evaluations. We replicated bot movements from a corpus of 500 games, sourced from the publicly accessible archives at https://rbc.jhuapl.edu/about. Subsequently, we replaced the original sensing method with our naive entropy sensing and made measurements throughout the games.

Blind Chess, sensing strategies are used to uncover information about the opponent's board state. For this purpose, we developed two entropy-based strategies: the Naive Entropy Sense and the Adapted Entropy Sense. Both strategies leverage the concept of entropy, a measure of uncertainty or randomness in data. In our scenario, entropy is computed based on the potential piece configurations on the board.

4.1. Naive Entropy Sensing Strategy

Our first strategy, the Naive Entropy (NE) sensing strategy, aims to minimize the entropy or uncertainty of the board state. Given a set of potential board states, the strategy calculates the entropy for each square on the board. This calculation is based on the frequency of each potential piece type (including an empty square) that could occupy a particular square across all possible boards. The entropy score for each square is then computed using the formula:

$$E = -\sum p(x) \log_2(p(x))$$

where $p(x)$ represents the probability of a certain piece being on the square. This probability is taken from the ratio of the pieces on the squares from all possible game states. The Naive Entropy Sense strategy then identifies the 3x3 region on the board with the highest aggregate entropy and selects the center square of this region for the sense action. This strategy effectively identifies the region with the highest level of uncertainty and gathers information to reduce this uncertainty.

4.2. Adapted Entropy Sensing Strategy

The Adapted Entropy (AE) sensing strategy builds upon the Naive Entropy strategy by adding further considerations. First, it includes a threat assessment element, where the potential threat posed by a square is calculated based on the pieces that could potentially attack it. The threat weight of each square is calculated by adding up the weights of the attacking pieces, where the weights correspond to traditional chess piece values.

Additionally, the AE strategy introduces a penalty for sensing squares that have already been sensed in previous turns. This penalty, computed with a decay factor, discourages the strategy from repeatedly sensing the same squares, thus promoting exploration of the board.

The AE strategy calculates the total score for each 3x3 region as a combination of the entropy score, the threat weight, and the sensed penalty. The region with the highest total score is selected for the sense action. This strategy aims to reduce the uncertainty of the board state while considering potential threats and past sensing actions to make a more informed decision.

5. Moving Strategy and Implementation

In the pursuit of determining optimal moves within the complex landscape of Reconnaissance Blind Chess, we employ the Sunfish engine (Ahle). This decision was prompted by Sunfish's advantageous lightweight architecture which, in contrast to more powerful engines such as Stockfish, allowed for greater flexibility in manipulating the source code to accommodate our specific requirements. Notably, we introduced modifications such as enabling the ability to castle, traversing through checks, and altering the termination condition of the underlying minimax algorithm from a checkmate state to the explicit capture of the king.

Our strategy for managing the inherent uncertainty in RBC involved the selection of n optimal and suboptimal board states, as evaluated by Sunfish's scoring algorithm. This algorithm incorporates a synthesis of centipawn values and piece-square-tables to quantitatively assess board states.

In scenarios where the estimated knowledge of the opponent is considered, a retrospective analysis for each of the n board states is performed. This process involves a replay from the initial board state to the current one, simulating the opponent's potential moves based on our known actions. A reasonable hypothesis is that a rational opponent will strive to minimize the overall entropy and pick an according sense. Subsequently, we average the scores of all possible board states from the opponent's perspective and integrate these into a composite ranking that encapsulates both our evaluation and the opponent's potential assessment.

The computed scores for each board state are then utilized to allocate computational resources for the search of the optimal move within a given board state. Board states with higher scores are apportioned more search time, implying their potential significance in determining the optimal move. We record all the candidate moves, along with their corresponding Sunfish scores, from each board state. The move with the highest cumulative score across all board states is chosen as our final move. This approach harmonizes our understanding of the game state with the potential understanding of our opponent, yielding a move that is robust to the inherent uncertainties of RBC.

6. Evaluation

We evaluate and compare two sensing strategies in this section. Figure 6 illustrates the comparison of the number of states eliminated in the game due to sensing. The NE strategy, which assumes that all legal moves are selected with equal probability, is designed to eliminate as many states as possible. However, in practice, it is observed that the AE strategy demonstrates a superior performance, eliminating a greater number of states in most turns. This observation shows that the strategy of observation is more than a simple mathematical computation of optimal solution but influenced by the game dynamics.

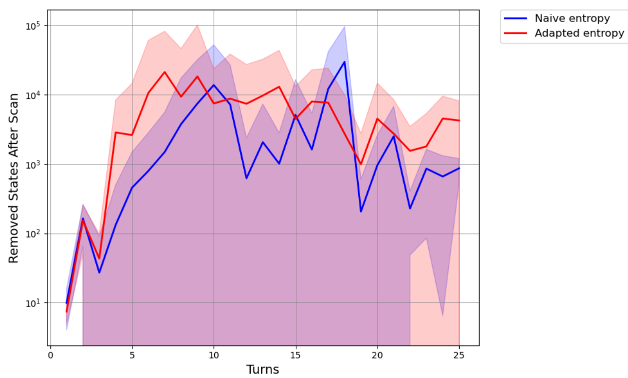

Figure 6. Number of States Eliminated Post-Sensing Using Various Strategies

	SunFish NE		SunFish AE	
	Basic	+ToM	Basic	+ToM
MinMax	20 / 20	20 / 20	20 / 20	20 / 20
Trout	14 / 20	17 / 20	16 / 20	18 / 20
StrangeFish	4 / 20	4 / 20	6 / 20	8 / 20

Table 1. Performance comparison of our agent against various opponent agents with or without Theory of Mind (ToM) by estimating opponents' knowledge.

Table 1 is the result of a summary games that were played against different agents under varied settings. For each combination, we execute 20 games. MinMax refers to a straightforward MinMax algorithm that selects the most preferred move for the highest-ranked boards. The Trout bot, which employs the Stockfish engine on a real-time estimation of the actual board state, serves as a standard bot for comparison. StrangeFish, currently the top bot on the official RBC leaderboard, was also included in the comparison. The evaluation was performed using both sensing strategies, with and without the supplementary estimation of opponents' knowledge.

The result shows that the AE strategy performs better than the NE strategy in general. This is consistent with our findings above. We also noticed that taking supplementary estimation of opponents' knowledge into account can improve the performance. The best performance comes from the bot using the SunFish engine with ToM. It is comparable with best existing bot, the StrangeFish. This shows how a deep understanding of knowledge in the game can guide the design of strategy that may result in better performance.

It is important to note that, as a work in progress, the results of our agent are presented without time constraints when estimating the opponents' knowledge. This means that we manually pause the game's timer for this specific calculation. With the current complexity of this estimation, the time required for its execution exceeds what would be feasible in a real game scenario. The optimization of our code remains a work in progress.

7. Discussion

Reconnaissance Blind Chess (RBC) poses a unique set of challenges due to the element of uncertainty inherent in the game. Unlike classical chess where the state of the board is fully observable, RBC adds an additional layer of complexity as players have only partial information about the board. In response to these challenges, we have explored various sensing and moving strategies designed to manage uncertainty and optimize decision-making.

Our exploration of sensing strategies centered on entropy-based approaches, specifically the Naive Entropy and Adapted Entropy strategies. These strategies aim to reduce uncertainty by selecting senses that provide maximum information. We found that the Adapted Entropy strategy, which incorporates consideration of potential threats and encourages exploration of the board, outperformed the Naive Entropy strategy. This observation suggests that information acquisition in RBC extends beyond mere sensory data and includes strategic elements such as understanding the opponent's potential moves. Furthermore, it could be intriguing to consider moving actions as another avenue for gaining information, instead of purely focusing on direct sensory inputs.

Complementing our sensing strategies, we also employed a moving strategy based on the Sunfish engine, which was further enriched by integrating our estimation of the opponent's knowledge. The decision to utilize a lightweight engine like Sunfish was motivated by the need for customization and flexibility to handle the unique characteristics of RBC. By leveraging both our understanding of the game state and potential assessment of our opponent's knowledge, we managed to construct a decision-making process that is more robust and tailored to the idiosyncrasies of RBC.

Despite these advancements, we acknowledge certain limitations and potential areas for improvement in our current approach. For instance, while considering the opponent's estimated knowledge has proven beneficial, it imposes computational demands that may affect real-time gameplay efficiency. In addition, our moving strategy could potentially be improved by incorporating more sophisticated threat assessment and defensive considerations.

The challenges posed by RBC align with the broader issues encountered in planning in partially observable environments. As such, we believe that strategies used in this domain, such as the Monte Carlo Tree Search for partially observable environments (POMCP) (Silver & Veness, 2010), the PEGASUS method (Ng, 2000), or the PFT-DPW algorithm (Sunberg & Kochenderfer, 2018), could be beneficially integrated into our approach. These algorithms, designed to manage uncertainty in complex environments, could enhance our decision-making process and consequently improve game performance.

While our focus so far has been on adapting and optimizing strategies within the framework of RBC, we recognize the potential to extend our approach to other contexts. For instance, logic-based approaches like Dynamic Epistemic Logic (DEL) (Van Ditmarsch et al., 2007) and Probabilistic Dynamic Epistemic Logic (PDEL) (Kooi, 2003) could be utilized to infer new knowledge in other imperfect information games. However, such an approach would need to address challenges related to the scale of the game.

Furthermore, we believe our approach could inspire the design of strategies in other settings where agents compete or cooperate with limited information. This includes other planning tasks, Q&A games, or scenarios where the acquisition of opponent knowledge is crucial. We are also interested in exploring the potential of machine learning algorithms, such as deep reinforcement learning, to learn optimal sensing policies and adapt to opponent behavior over time.

Exploring the behavioral patterns of our opponent presents another exciting avenue for enhancing our estimation of the true board state. Similar to classical chess where players can often be categorized into distinct types (Kaehler, 2022), it would be interesting to investigate if artificial intelligence opponents in RBC exhibit specific styles of play. If such categorizations can be discerned, it would allow for more refined estimations of the opponent's strategy, ultimately improving our decision-making process and performance in the game.

In conclusion, our study underscores the importance of robust sensing and moving strategies in managing uncertainty in RBC. Our entropy-based sensing strategies and Sunfish-based moving strategy, inspired by Theory of Mind (ToM), provide a strong foundation upon which future work can build. We encourage further exploration of advanced algorithms and techniques, such as probabilistic reasoning and machine learning, to address the limitations of our current approach and deepen our understanding of game dynamics in RBC and other similar environments. We anticipate that these research directions will yield exciting advancements in the field, and we look forward to seeing how they develop in the coming years.

References

Ahle, T. D. GitHub - thomasahle/sunfish: Sunfish: a Python Chess Engine in 111 lines of code. https://github.com/thomasahle/sunfish. [Accessed 24-May-2023].

Aru, J., Labash, A., Corcoll, O., and Vicente, R. Mind the gap: Challenges of deep learning approaches to theory of mind. *arXiv preprint arXiv:2305.11111*, 2023.

Carlson, S. M., Koenig, M. A., and Harms, M. B. Theory of mind. *Wiley Interdisciplinary Reviews: Cognitive Science*, 4(4):391–402, 2013.

Carruthers, P. and Smith, P. K. (eds.). *Theories of Theories of Mind*. Cambridge University Press, 1996.

Kaehler, A. Player types: The pragmatic, 2022. URL https://en.chessbase.com/post/player-types-the-pragmatic. Accessed on June 27, 2023.

Kooi, B. P. Probabilistic dynamic epistemic logic. *Journal of Logic, Language and Information*, 12:381–408, 2003.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Ng, A. Y. Pegasus: A policy search method for large mdps and pomdps. In *In Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pp. 406–415, 2000.

Silver, D. and Veness, J. Monte-carlo planning in large pomdps. In *Advances in Neural Information Processing Systems*, volume 23, 2010. URL https://proceedings.neurips.cc/paper_files/paper/2010/file/edfbe1afcf9246bb0d40eb4d8027d90f-Paper.pdf.

Sunberg, Z. N. and Kochenderfer, M. J. Online algorithms for pomdps with continuous state, action, and observation spaces. *Aeronautics and Astronautics Dept., Stanford University*, 2018. URL https://arxiv.org/abs/1709.06196.

Van Ditmarsch, H., van Der Hoek, W., and Kooi, B. *Dynamic epistemic logic*, volume 337. Springer Science & Business Media, 2007.