How does using Theory of Mind in an Agent-Based Model Influence the Results of the Two-Player Board Game of Onitama

Bachelor's Project Thesis

Teun Boekholt, s4716515, t.boekholt@student.rug.nl,
Supervisor: Dr H.A. De Weerd

Abstract: The subject of theory of mind has come up in a lot of modern research, but mainly in regard to relatively simple games. In this study a different approach is taken, where theory of mind functionality is added upon an existing minimax algorithm with alpha beta pruning for the two-player board game of Onitama. A possible advantage of theory of mind could hereby be found for more complex games. However, in this study the addition of theory of mind did not prove to provide its user with an advantage. Further research into the usage of theory of mind in more complex game settings is necessary to better understand the effectiveness of theory of mind.

1 Introduction

Theory of mind (Premack & Woodruff, 1978) has extensively been studied in numerous papers (see, e.g., Rabinowitz et al. 2018 for an overview). However, these investigations often rely on relatively simple zero-order models. In this paper we look at the advantage of theory of mind in the two-player strategic game Onitama, using a sophisticated zero-order model.

Previously, agent-based modelling has been used to analyze theory of mind in the game of rock-paper-scissors and variations of this game like the Mod game (De Weerd et al., 2013; De Weerd et al., 2014), but also more social games like Werewolves (Aylett et al., 2014). Since in real-world multi-agents environments individual agents often have to work together, studies in which cooperative games were modelled should not go overlooked. For example, the benchmark cooperative card game Hanabi has extensively been subjected to numerous agent-based models using theory of mind (Lerer et al., 2020; Dupuis, 2022; Bard et al., 2020).

Before going into more detail about what type of model would be the right approach for the game of Onitama, we will first delve into the concepts of theory of mind and agent-based modelling.

Theory of mind is the ability to attribute mental states to people outside of ourselves. We all have our own beliefs, desires and intentions, but to actively reason about the mental contents of other people is what we call theory of mind. Theory of mind can be used on many levels, so-called orders. The higher the order of theory of mind, the deeper the agent reasons about the mental states of other agents. Zero-order theory of mind (ToM_0) is the lowest possible level of theory of mind. ToM_0 does not involve reasoning about other people's mental states. However, this does not mean one does not take the actions of other's into account. For example, in the game of rock-paper-scissors a ToM_0-agent can still discover that playing rock against someone who often plays scissors is the superior strategy (De Weerd et al., 2013). For this the ToM_0-agent does not need to reason about the other agent's beliefs and intentions. A first-order theory of mind (ToM_1) agent expands on this by also reasoning about the decision-making process of other agents. For example, in rock, paper, scissors this would mean that a ToM_1-agent first thinks about what the other agent would do based on the ToM_1-agent's own actions. It can do this by reasoning as a ToM_0-agent from the other agent's perspective. For example, if the ToM_1 agent knows that they themselves have played rock for the past three games, they reason from the perspective of

their opponent to reach the conclusion that their opponent will most probably play paper to counter the rock. Based on this prediction from the perspective of the opponent the ToM_1 will play scissors. A first-order theory of mind (ToM_1) agent 'puts themselves in the shoes' of another agent. When we think about a ToM_2-agent, you could say that a they even go a step further by first putting themselves in the shoes of another agent, and after that putting themselves in their own shoes again from the viewpoint of that other agent. This continues so on for theory of mind agents at higher levels.

According to the Machiavellian Intelligence Hypothesis as described by Byrne & Whiten (1988), individuals that make use of higher orders of theory of mind are equipped with an advantage in the evolutionary process. De Weerd et al. concluded in 2013 that this is not always the case, as they found that in the game of rock-paper-scissors, ToM_4-agents did not have an advantage against ToM_3-agents. Still, the main result was that an advantage was present for ToM_1 and ToM_2 agents. In the same paper by De Weerd et al. another relatively simple game (although more advanced than rock, paper, scissors) called limited bidding was also put to the test, which yielded the same results as the rock-paper-scissors game.

The benefits of ToM have thus far (in the studies mentioned above) mainly been investigated using "toy models" and with the use of ad-hoc ToM_0 models. It is interesting to see whether benefits exist on top of sophisticated ToM_0 models, since it is necessary that this pattern is also prevalent in more complex games in order to effectively support (or challenge) the Machiavellian Intelligence hypothesis. To obtain a better understanding of the benefits of higher-order theory of mind reasoning, we therefore take a different approach in this paper. Rather than using a simple game, we investigate the game of Onitama.

To simulate the game of Onitama, agent-based modeling (ABM) will be used. ABM is the process of modeling a complex scenario by simulating individual agents and how they interact with their environment using actions. In this scenario the agents will be the players, the environment will be the game and the actions will be the different moves with which the players can influence their environment. In the game of Onitama, moves are made by playing cards. Once a card corresponding to a move is played, the other player will get to use this card in their next turn. Playing a strong card therefore does not only come with the downside of losing that card, but also provides your opponent with that same card. This same principle can be applied to many real-world scenario's, like economics and political decision-making. By using the game of Onitama, I hope to simulate a scenario that is more likely to imitate real-world decision-making processes.

The remainder of this paper is structured as follows. We will first explore the game of Onitama and how it can be implemented as an agent-based model in Sections 2.1 and 2.2. Then the zero-order model will be described in detail (Section 2.3.1), after which we elaborate on the higher orders of theory of mind used in this study (ToM_1 and ToM_2) in Sections 2.3.2, 2.3.3 and 2.3.4. After running some experiments using these models, the results are presented and analyzed in Section 3. From this a conclusion is drawn (Section 4) and several pecularities are discussed (Section 5).

2 Methods

2.1 Rules of Onitama

Onitama is a two-player strategic board game that was designed by Shimpei Satio and got published in 2014 by Arcane Wonders. Thematically the game entails controlling a group of martial artists that need to outmaneuver each other to prove their superiority.

The game is played on a 5x5 square grid, with each player controlling five pieces. Four of them are so called 'Students', and one the 'Master'. The pieces are initially placed on the board as shown in Figure 2.1, with the Masters residing on the 'Temple Arch'-square (from now on simply referred to as 'Temple'). Next to this, all the cards are shuffled, each player receives two cards, and an additional one is placed in the middle. The rest of the cards are not used for this game. Thus only five cards are used each game.

The goal of the game is to either capture the opponent's Master piece, or to reach the opponent's Temple with your own Master. This is done by playing cards. When it is a player's turn, they must play one of the two cards in front of them to make

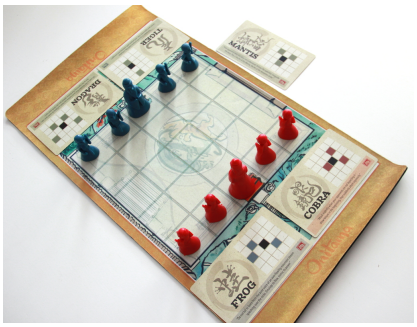

Figure 2.1: The setup of a game of Onitama. The Masters are placed on the middle squares (the Temples) on both sides of the board, flanked by their students. Each player receives two cards, and one card is played in the middle.

a move with one of their pieces. For example, with the 'Frog' card shown in Figure 2.2, the player can move one of their pieces either one square diagonally to the top left or bottom right relative to their current position. Or the piece can move two squares to the left. Pieces cannot land outside of the grid or on pieces of their own color. They can, however, jump over all pieces (like the knight in chess). When a piece lands on a piece of the enemy's color, the enemy piece is captured, removed from the board and the piece that captured it is now placed on that square. If the captured piece was the Master, the player that captured the piece has won the game.

Figure 2.2: One of the possible playing cards in the game of Onitama. It shows all the possible moves that can be made using this card.

After a player's turn they put the card they just played in the middle to the side of the board. They take the card that was already laying at the side of the board as their new second card. When the other player played their turn they follow the same procedure, and thus the five cards constantly rotate between the two players.

The game continues until either a Master piece is captured, or as soon as a player manages to place their Master on the opponent's Temple square.

2.2 Implementation of Onitama

To implement Onitama into a virtual environment where simulations with different agents could be run, the third-party model by Weiner (2018) was used. This model contained a lot of the basic functionality needed to play the game of Onitama. In Figure 2.3 you can see the user interface of this model.

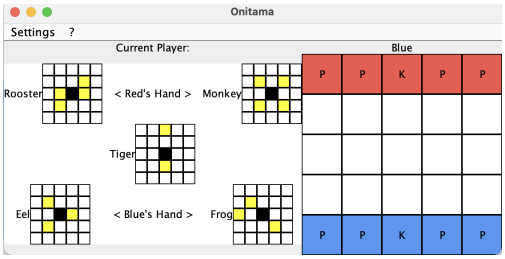

Figure 2.3: General user interface of the Onitama board game as implemented by Weiner (2018).

With this model, you can play with two humans, against an AI player, or let two AI players play against each other. The computer-controlled agents could be set to three different difficulty levels: easy, medium and hard. The AI model makes its moves using a basic search tree algorithm which utilizes the minimax algorithm with alpha-beta pruning (see Section 2.3.1). The difficulty of the AI agents was determined by the maximum depth that they explored in the search tree (respectively 3, 4, and 5).

For the purpose of this study, the model was further expanded to include simulation functionality. With this function, two ToM-agents could be made to play a predefined number of games against each other, after which the amount of wins for both players was recorded. This made it easy to check which agent had the upper hand.

In the initial model all the cards from the basic game were used. At the start of each game the deck is shuffled and five random cards are used for each round. However, some combinations of cards will lead to the model getting stuck in a loop, where

the same sequence of moves is played over and over again. Why this happens is explained in Section 2.3.1. After some experimentation with several computer-controlled agents (at varying minimax depths) playing against each other, a selection of five cards was found that did not lead to a stalemate situation. And thus these cards are used throughout the rest of this research. The five cards are the *tiger*, *frog*, *rooster*, *monkey* and *dragon*. The corresponding moves are shown in Figure 2.4.

Figure 2.4: The five cards used for the model in this paper.

2.3 Theory of Mind Models

2.3.1 Zero-Order Theory of Mind (ToM_0) & Minimax Algorithm

Zero-order theory of mind agents (ToM_0) are not capable of using theory of mind functionality. They do not reason about the intentions and decision-making processes of their opponent and therefore have no real use for the prediction values as described in Section 2.3.2. ToM_0-agents base their actions solely on a standard minimax with alpha-beta pruning algorithm.

The minimax algorithm can be represented as a search tree (as shown in Figure 2.5), where an agent goes down the branch that yields the largest utility value. Nodes represent states of the game, and the color (blue or red) of the node indicates whose turn it is. The example in Figure 2.5 is based on a two-player game where each player can only take two actions during their turn. The bottom four nodes' values match the utility values for the blue player. So when both players play optimally, the blue player always goes down the branch that yields the highest utility, and the red player acts vice versa. This is why 20 is the highest possible utility value that can be achieved for the blue player.

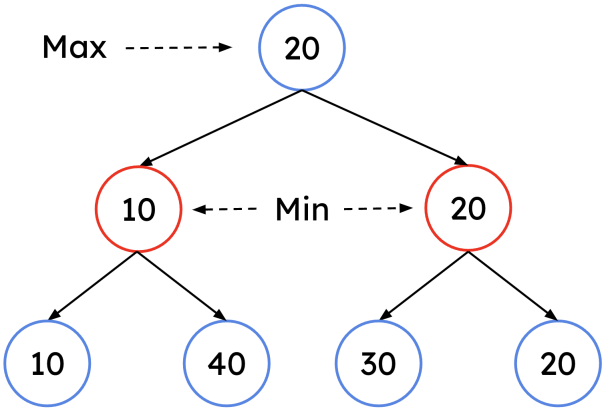

Figure 2.5: An example search tree of the minimax algorithm.

To get the earlier mentioned utility value, the current state of the game is evaluated using an evaluation function. Below is the evaluation function used in this study, which was directly adapted from the code by Weiner. Here the values with *cur* in front of them correspond with values of the current player's piece count and distance from their master to the opponent's temple square, whilst values with off in front of them correspond with values of the off player's piece count and distance from their master to the current player's temple square.

$evalutionscore = (curCount - curCloseness) - (offCount - offCloseness)$

In this case, the game state is evaluated based on two factors. One is the remaining pieces on the game board ($curCount$ and $offCount$). If it is the blue player's turn, blue pieces yield a positive utility function of simply 1 per piece, whilst red pieces contribute negatively. Secondly, the distance between a player's master piece and the opponent's temple square is measured using a simple Euclidian distance function $((x_m - x_t)^2 + (y_m - y_t)^2)$, where x_m and y_m are the coordinates of the current player's master, and x_t and y_t of the op-

posing player's temple). These correspond to the $curCloseness$ and $offCloseness$ values in the formula. The closer the master is to the temple square, the higher the utility value will be.

It is easy to imagine that Onitama's search tree would be much larger than the one in Figure 2.5. Every turn a player can pick between two cards, which both allow for an average of 3.6 possible moves, which can be applied to five playing pieces. This amounts to one node expanding into $2 \cdot 3.6 \cdot 5 = 36$ nodes. With a branching factor of 36, each level deeper into the tree would require the algorithm to go over 36^k nodes for depth k. For a depth of only 3 (the lowest depth used in this study) this would already amount to just shy of 50.000 nodes. To go over all these nodes would be too inefficient, hence why alpha-beta pruning is used.

Alpha-beta pruning is a method that is commonly used to limit the search space of the minimax algorithm (Russell & Norvig, 2016). Branches of the search tree that are not worth exploring are 'pruned'. This is done by keeping track of two extra values, α and β. The α value represents the highest utility that the player trying to maximize the utility (the blue player in Figure 2.5) has found so far, and the β value represents the lowest utility that the player trying to minimize the utility (the red player in Figure 2.5) has found so far. If at any point the utility at the current node becomes lower than α for the maximizing player or higher than β for the minimizing player, the branch is not further explored since there will not be a utility value that can help the players in making a decision. A branch can also be cut off if α ever is equal or greater than β. This means the branch is not worth exploring, as it will not influence the final decision made by the maximizing player.

As mentioned in Section 2.2, the agents could get stuck in a loop with a certain setup of cards. When this happens, both agents continue playing the same sequence of moves and the game essentially reaches a stalemate where the same state of the game board is visited over and over again. Since all information is open and there is no randomness involved with the minimax algorithm, a certain position of pieces and cards will always yield the same optimal move when running the minimax algorithm at the same depth. Hence, it can happen that if our hypothetical blue player starts playing and both players have played two moves, the board will be in the same state as it was at the beginning of blue's first turn. Thus blue will play the same move again and the game reaches a looped state. Luckily a combination of cards was found which did not lead to these types of scenarios, as mentioned in Section 2.2.

In summary, zero-order theory of mind agents always make their moves based solely on the previously described minimax with alpha-beta pruning algorithm at a depth of three. In the model used for this study, ToM_0-agents actually were able to make predictions about the depth that other agents were playing at. This was done to warrant the simplicity of the model's implementation. Since they themselves cannot use the minimax algorithm at a depth higher than three, the use of predictions about their opponent's beliefs and intentions is inconsequential. They essentially function in the same way as an agent with a predefined depth of 3.

2.3.2 Higher Orders of Theory of Mind

Before going into more detail regarding the separate orders of theory of mind, we will first look at the general implementation of theory of mind in this study. The ToM architecture in this study was greatly inspired by the formulation in De Weerd et al. (2013).

As each game is initialized, all theory of mind agents are equipped with three prediction values: d_2, d_3 and d_4. These prediction values can each take a value between 0 and 1 and represent the confidence of an agent regarding the depth d at which their opponent uses the minimax algorithm, as was explained in Section 2.3.1. If $d_2 = 0.2$, $d_3 = 0.2$ and $d_4 = 0.8$, the agent believes that their opponent is most likely an agent that reasons at a depth of 4. Note that the prediction values do not necessarily add up to a value of 1. They are merely used as a hierarchical indication of what depth an agent believes its opponent to reason at. Also note that the prediction values do not correspond directly with the real maximum depths of the ToM-agents. For the ToM_0-agent, ToM_1-agent and the ToM_2-agent the maximum reasoning depths respectively are 3, 4 and 5, whilst the prediction values are used to predict one's opponent to reason at depth 2, 3 or 4. This is done because agents will always try to reason at a depth which is one higher than that of their opponent (as will be described later in this

Section), and therefore it is necessary that a d_2 exists. Otherwise agents would never choose to reason at depth 3. Furthermore, a d_5 would be futile, since no agents used in this study can reason at a depth of 6. Please keep this design choice in mind when reading through the rest of this Section.

Every turn, after an agent has performed their own action (a_i) utilizing a minimax algorithm with alpha-beta pruning (as was described in more detail in Section 2.3.1) and depth n, they make a prediction ($\hat{a}_j$) about what action they believe the other player is going to take. They choose $n = i + 1$ for an i that corresponds with the highest d_i value. If there is a tie for the highest prediction value, the d_i is chosen with the lowest i. This means that reasoning at lower depths is preferred when there is a tie. For example, if d_3 is equal to d_4, the agent chooses to use a depth three model for their opponent. The logical support for this design choice is that when two models are both supposed to be the best, the one that is most efficient to use (hence the simpler model) should be chosen. To make the prediction a_j about the opponent's move, they will again use the same minimax algorithm they used for determining their own move, but they will now reason at depth i. This means that they predict their opponent to reason at depth i as well. Hence why they choose i according to the highest d_i prediction value. Then, next turn, after the opponent has performed their action a_j, the agent will compare the a_j to the prediction they made in their own turn ($\hat{a}_j$). The prediction value that was used for making the prediction is then updated according to the formula below. This means that only the highest prediction value d_i is updated according to this formula. The update formula for the other two prediction values is described below. This formula is the same formula that is used for updating the confidence levels in the study by De Weerd et al. (2013).

$$d_k := \begin{cases} \lambda + (1 - \lambda) \cdot d_k & a_j = \hat{a}_j \\ (1 - \lambda) \cdot d_k & a_j \neq \hat{a}_j \end{cases}$$

Here λ represents the learning speed, which can be a value between 0 and 1, and impacts how quickly agents adjust their prediction values. For the experiments done in this study λ was set to 0.6.

If $a_j \neq \hat{a}_j$, along with the highest prediction value declining, the other two prediction values increase with the learning speed, following the formula $d_k = d_k + \lambda$. As mentioned earlier, this means that the update function is different for the two prediction values that were not used for making the prediction than the update function that was used for the highest prediction value. For example, in the scenario where $d_2 = 0.8$, $d_3 = 0.2$ and $d_4 = 0.2$, the agent will believe their opponent to be reasoning at a depth of 2, as d_2 is the highest prediction value. So first, the agent will play at their own move using the minimax algorithm at a depth of 3, as this is one higher than they believe their opponent to be. The agent then makes a single prediction about what move their opponent is going to play. This prediction is based on the model they believe their opponent to be. In the case where d_2 is the highest, a model of depth 2 will be used for making this prediction. In the end, the agent will have made only one prediction about the action of their opponent, which is an innovation on already existing work by De Weerd et al. (2013), where a prediction is made based on all an agent's models of their opponent. If it turns out that the prediction the agent made (assuming that their opponent reasoned as a depth 2 agent) was incorrect, d_2 will be adjusted to 0.32 ($(1 - \lambda) \cdot d_2$), and the other prediction values will be increased to 0.8 ($d_k + \lambda$).

Again, it should be noted that the way in which prediction values are used in this study substantially differs from how the confidence levels (which could be understood as the same concept as the prediction values) are used in the studies by De Weerd et al. that were mentioned in the introduction. Whereas De Weerd et al. use confidence levels to represent an agent's confidence in its own order of theory of mind, in this study prediction values are used to represent an agent's confidence in the order of theory of mind/depth of their opponent. To understand the core of this project, it is important to be aware of this difference and what it entails. For example, in the studies of De Weerd et al. c_1 indicated an agent's confidence in its own first order theory of mind model, which was used to discern ToM_0 opponents. c_1 could therefore also be seen as a representation of the confidence in the opponent being a ToM_0-agent. However, in this study d_3 is used to indicate the same thing. The prediction values correspond directly with the opponents' order of theory of mind/depth, instead of with an agent's own theory of mind models.

How each separate type of ToM-agent utilizes the prediction values to make accurate predictions and adjust their own course of actions accordingly is clarified in the Sections below.

2.3.3 First-Order Theory of Mind (ToM_1)

First-order theory of mind agents have the ability to utilize the alpha-beta pruning minimax algorithm at a depth of either three or four. To decide at which depth they will use the algorithm, they make use of the prediction values as described at the start in Section 2.3.2. After a ToM_1-agent has decided on what move to make, they make a prediction of what they think their opponent is going to play after them.

The agents in this study can only discern the depth of their opponent if their opponent uses a lower order of theory of mind than the agent itself. For example, ToM_1-agents can only really discover whether their opponent is a ToM_0-agent or not. First-order theory of mind agents are not able to make a distinction between other ToM_1-agents and ToM_2-agents.

ToM_1-agents and ToM_2-agents make the predictions of their opponents' actions up to a maximum depth n_j that is equal to their own maximum depth (n_i) minus 1. Thus $n_j \leq n_i - 1$. This feature is present in the model to ensure the consistency of predictions about the opponent's depth. If an agent would be able to make predictions at the same depth as what they play their own moves at, they would essentially use a different prediction model for their opponent when using the minimax algorithm.

The problem that would arise if both the prediction about the opponent's move and the decision about an agent's own move are made using the same depth is made clear in Figure 2.6. Here the squares with a P in it represent a turn of the current player/agent, whilst the squares with an O in it represent the moves of the current player's opponent. The illustration should be seen as a point of view of the current player as it looks into the future (using the minimax algorithm) at a specific depth (four in this case), to decide on both its own optimal move and the most probable move of its opponent. As can be seen in Figure 2.6, in the case that the same depth is used, the prediction of the opponent's move is first made at an effective depth

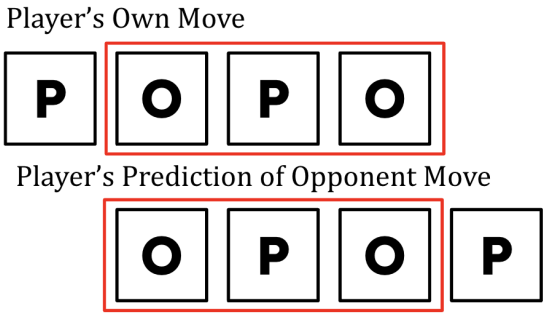

Figure 2.6: An example of why the prediction depth n_j should be one lower than the depth n_i used for an agent's own move.

of *three* (as indicated by the red outline in the upper part of the illustration), after which a depth of *four* is used to make the actual prediction of the opponent's move. The red outline in Figure 2.6 shows that these two predictions do no match. Therefore the prediction of the opponent's move is made at a maximum depth which is one lower than the maximum depth used for deciding on a player's own move. In this way the P-square in Figure 2.6 that falls outside of the red outline is not taken into account and thus the two predictions will match.

In summary, ToM_1-agents can reason using a depth of 3 or 4, and can adjust their depth to the accuracy of the predictions they make about the opponent's behavior. They can only distinguish a ToM_0/depth 3 agent.

2.3.4 Second-Order Theory of Mind (ToM_2)

Second order theory of mind agents, also the highest order theory of mind agents that will be covered in this study, are able to reason at depths of three, four and five. Essentially their implementation is exactly the same as ToM_1-agents. The only difference is that ToM_2-agents are able to use the minimax algorithm at a depth of five. This also means that the same issue arises that was illustrated in Figure 2.6. Fortunately the same solution to the problem (limiting the maximum depth that is used when making a prediction about the opponent's move) also proved effective for ToM_2-agents.

To decide at what depth a ToM_2-agent will use the minimax algorithm, they see whether their d_2,

d_3 or d_4 is higher. Depending on which prediction value has a greater value, the ToM_2-agent will respectively reason at a depth of 3, 4 or 5.

Just like ToM_1-agents are not able to discern other ToM_1-agents, ToM_2-agents are not able to discern other ToM_2-agents. This is the case because agents can only make predictions about their opponent at a depth that is one lower than their own maximum depth, as described in Section 2.3.2. What impact this has on the game play and results will be further explored in Section 3.

2.3.5 The difference between ToM and minimax

To an attentive reader the question may have arisen what exactly the difference is between theory of mind and the minimax algorithm. This is a very interesting question. After all, the minimax algorithm essentially is a form of theory of mind in itself. Making predictions about what the opponent is going to do by reasoning like your opponent precisely aligns with the definition of theory of mind given in Section 1. However, in this study ToM is used to show a possible advantage that ToM brings to the competition in a varying way. Here, the term ToM specifically relates to the ability of ToM-agents to not only use the minimax algorithm, but to also try to work out what type/depth of minimax algorithm their opponent is using to adjust their own depth accordingly. It is true that minimax resembles ToM in a lot of ways, but the ToM mechanism described in the past section can be seen as an extra type of theory of mind.

Now one might be left to wonder why this addition should prove to provide an advantage. The supposed advantage is that the issue as illustrated with Figure 2.6 would be circumvented. This type of ToM should ensure that agents use the minimax algorithm in a way that is consistent with the depth of their opponent.

3 Results

After having completed the full implementation of Onitama and theory of mind (ToM) agents that can play the game according to the mechanisms described in Section 2.3, the experiment was run. Several simulations were run where varying combinations of ToM-agents played 2000 games against each other. The same type of simulation was run for agents that did not use theory of mind, but only used the minimax algorithm at predefined depth (3, 4 or 5).

Percentage of Games won for Agents of Equal Depths and ToM		
Depths	Agent 1	Agent 2
Depth 3 vs. 3	49.75%	50.25%
ToM_0 vs. ToM_0	49.05%	50.95%
Depth 4 vs. 4	50.10%	49.90%
ToM_1 vs. ToM_1	51.35%	48.65%
Depth 5 vs. 5	49.15%	50.85%
ToM_2 vs. ToM_2	50.50%	49.50%

Table 3.1: The results of 2000 games played between agents of the same depth or ToM-order

In Table 3.1 the results of agents of the same depth/ToM-order can be found. It should be noted that the depth 3 agent and the ToM_0-agent are essentially the same type of agent, as they can both only reason at depth three. It should be expected that both agents in all of these games win more or less 50% of the games played against each other. However, since the starting player is not determined at random (but rather is dependent on the card that starts in the middle), there exists a possibility that a certain combination of cards favors a certain player at specific depths/ToM-orders. To circumvent this potential issue, both agents varied between playing as the blue player for the first 1000 games, and then as the red player for the last 1000 games. For all the results presented in this section this manner of simulation was chosen when pitting two agents against each other.

After this method of running the experiment was applied, the expected results emerged, as can be seen in Table 3.1 and Figure 3.1. Agents of similar depths and ToM-levels are an equal match. None of the results in Table 3.1 are significant, according to a standard Z-test (proportion test), as none of the p-values were lower than the significance level of 0.05. The exact p-values and Z-statistics can be found in the Appendix.

The results in Table 3.2 were once again acquired from running 2000 games, where both agents switched color after 1000 games. For example, in

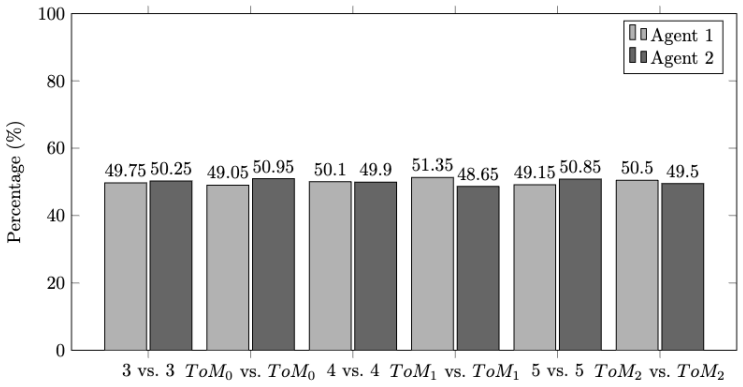

Figure 3.1: The win percentages after a variety of agents played 2000 games against each other.

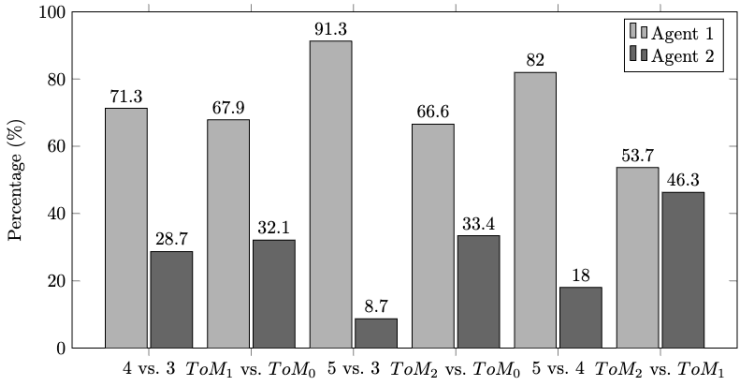

Figure 3.2: The win percentages after a variety of agents played 2000 games against each other.

the case where the ToM_0-agent played against the ToM_1-agent, for the first 1000 games the ToM_1-agent played as the red player and the ToM_0-agent as the blue player.

It gets more interesting once we start to look at the results that were produced by pitting agents of differing depths and ToM-levels against each other, as can be seen in Table 3.2 and Figure 3.2. Significance is indicated with a star in Table 3.2. Exact p-values and Z-statistics can again be found in the Appendix. It becomes apparent that higher orders of theory of mind and greater depths provide an advantage against agents that reason at lower depths and ToM-levels. However, the difference in win rates is significantly (see p-values in the Appendix) larger for agents that have a greater depth than their opponent. For example, an agent that reasons of depth 5 wins 82% of the games they play against their opponent that reasons at a depth of 4. In contrast, a ToM_2-agent (which also can reason at a maximum depth of 5 as described in Section 2.3.4) wins only 53.70% of the games against a ToM_1 opponent.

When letting ToM-agents play against agents that reason at a predefined depth, the results of which (including statistical significance) can be seen in Table 3.3 and Figure 3.3, a few particularly interesting results comes to light. The first is that ToM-agents seem to not only lack an advantage against their more restricted opponents that reason at a set depth, but in some scenarios are even clearly at a disadvantage. The most peculiar result probably comes from the fact that a depth 4 agents proves to do better against a ToM_2-agent than against a ToM_1-agent, whilst this is quite counter intuitive. However, it should be noted that

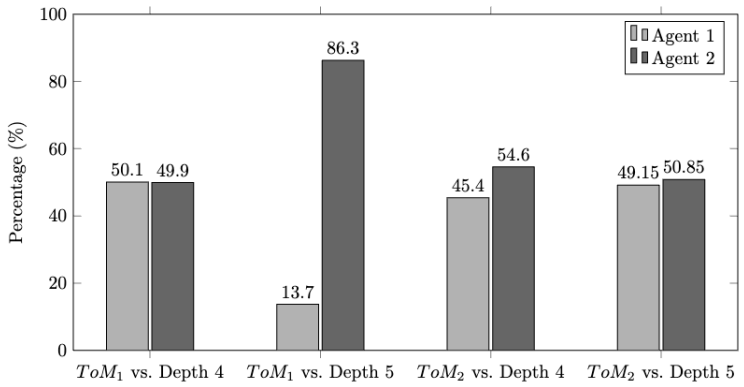

Figure 3.3: The win percentages after a variety of agents played 2000 games against each other.

Percentage of Games won for Agents of Differing Depths and ToM		
Depths & ToM-orders	Agent 1	Agent 2
Depth 4 vs. 3*	71.30%	28.70%
ToM_1 vs. ToM_0*	67.90%	32.10%
Depth 5 vs. 3*	91.30%	8.70%
ToM_2 vs. ToM_0*	66.60%	33.40%
Depth 5 vs. 4*	82.00%	18.00%
ToM_2 vs. ToM_1*	53.70%	46.30%

Table 3.2: The results of 2000 games played between agents of differing depths and ToM-orders. Significant results are marked with a *, meaning the p-value is below the significance level of 0.05.

Percentage of Games won for Agents of Differing Depths and ToM		
Depths & ToM-orders	Agent 1	Agent 2
ToM_1 vs. Depth 4	50.10%	49.90%
ToM_1 vs. Depth 5*	13.70%	86.30%
ToM_2 vs. Depth 4*	45.40%	54.60%
ToM_2 vs. Depth 5	49.15%	50.85%

Table 3.3: The results of 2000 games played between agents of differing depths and ToM-orders. Significant results are marked with a *, meaning the p-value is below the significance level of 0.05.

there is a large difference between the amount of games the ToM_2-agent won as the red player and the amount of games they won as the blue player. Namely, as the red player they won 590 of the 1000 games (59%), whilst as the blue player they only won 318 of the 1000 games (31.8%), which is a considerable difference that was not seen for any other pairing of agents. No clear explanation could be found for why this pairing is so colour dependent.

4 Conclusions

Following the results that were acquired in Section 3, no clear evidence could be found which proves that ToM-agents (as they are implemented in this study) are at an advantage when playing the two-player board game of Onitama in respect to agents that reason at a predefined depth. If anything, using theory of mind proved to be a disadvantage in some cases.

Especially ToM-agents that have a maximum depth that is higher than that of their non-ToM opponent would be better off if they just constantly reasoned at their maximum depth. For example, a ToM_2-agent (that has a maximum depth of 5) wins only 45.40% of the games against an agent that reasons at a predefined depth of 4, whilst a regular depth 5 agent that does not used theory of mind wins 82.00% of their games against that same depth 4 opponent.

One interesting finding is that a ToM_1-agent performs slightly better against a depth $3/ToM_0$-agent (they function in essentially the same manner) than a ToM_2-agent performs against this type of opponent. It should be noted that both the ToM_1-

agent and the ToM_2-agent do not perform as well as their predefined depth counterparts that reason at depths 4 and 5 respectively. However, the depth 5 agent does perform significantly better against a depth 3 agent than a depth 4 agent does. This contrast with ToM-agents is quite interesting, and at a first glance it could be explained by the fact that a ToM agent that reasons at a depth of 5 (such as a ToM_2-agent) uses a model of depth 4 to predict the behavior of the opponent, whilst this is not correct for a depth 3 agent. For a more detailed explanation about how the agents used in this study reason you can read Section 2.3. However, if this was the case then it would also be expected that a regular depth 4 agent also performs better against a depth 3 agent than a depth 5 agent does, but this is not the case. Therefore the precise reason for this anomaly remains unknown, although there is one possible explanation. As was explained in Section 3, the colour dependency was extraordinarily high when a ToM_2-agent played against a regular depth 4 agent. Why this phenomenon only occurred here could not be explained with certainty. One possibility is that there is a specific starting configuration of the five cards that grants the depth 4 agent a quick or hard to counter winning combination of moves when playing as the red player, which would explain the low win rate of 31.80% for the ToM_2-agent when playing as the blue player.

Furthermore, ToM-agents with a maximum depth n that played against opponents with a set depth of n had about the same win chance. This is a further indication that ToM does not provide an advantage in this particular game, and with this particular implementation. There was not one case that could be found where ToM-agents had an edge over their simpler opponents. The hypothesis that the usage of theory of mind in the two-player board game of Onitama would provide an advantage is therefore not supported by the results of this study.

Possible explanations and faults of the approach that was taken, along with what these results say about the usage of theory of mind in complex games and possibilities for future research will be explored in the next Section.

5 Discussion

Whereas in the research of De Weerd et al. (2013) theory of mind did prove to be a significant advantage in more simple games, in this study theory of mind did not have such an impact. Let us explore why this might be the case.

But first it must be stated that in the end, the effectiveness of theory of mind really comes down to what we define as theory of mind. There are many possible ways to go about inserting theory of mind in an agent-based model. As described in Section 2.3.5, in this study the theory of mind aspect was implemented as an addition on the minimax algorithm, by giving agents the possibility to alter the depth at which they use the algorithm. However, one could also make a solid argument for the case that the minimax algorithm is a form of theory of mind in itself (as was also briefly discussed in Section 2.3.5). In this case, theory of mind would have a considerable impact/advantage in the game of Onitama. For now, however, let us focus on why theory of mind does not have an impact in the way that it was implemented in this study.

One of the possible reasons might be that the evaluation function of the minimax algorithm is incorrect. If it were correct, the expectation would be that a ToM_2-agent plays better against a depth 3 agent than an agent that reasons at a predefined depth of 5 will play against that same agent, because the ToM_2-agent will adjust its depth to level 4 if playing an opponent of depth 3. This may seem counter intuitive, but since a ToM_2-agent will reason at depth 4, they use a more accurate model of the opponent's behavior. If, however, the evaluation function in the minimax algorithm (which determines the utility function for each game state) is specified in a wrong manner this could impede the workings of the algorithm. Currently, the evaluation function only considers the amount of pieces on the board and the proximity to the opponent's temple square. Since there are no cards that allow a piece to move one square in the forward direction, a possible flaw immediately comes to light. For example, if a player's master pawn is one square in front of the opponent's temple but has no card to reach it, this game state still receives a higher utility value than if a player's master would be two spaces away but could reach the temple in one move with the *tiger* card. A faulty evaluation function

might therefore lead to the malfunctioning of the entire model.

It could also very well be the case that adding theory of mind functionality on top of an already existing minimax algorithm is simply not beneficial. However, to further support this theory more research that involves using theory of mind in complex game settings needs to be done. If one were to do another study regarding the game of Onitama, a different evaluation function should probably be used. Different configurations of cards could also be taken into account. Along with this, theory of mind could be implemented in a different way altogether. There are a lot of ways to go about this, but for one, a closer resemblance could be held with the research by De Weerd et al. (2013) from which this study took a lot of inspiration. Theory of mind should also be explored regarding other complex game settings.

All in all, the Machiavellian Intelligence Hypothesis as described by Byrne & Whiten (1988) is not supported by this study. Using theory of mind does, according to the experiments performed, not provide an advantage in the two-player board game of Onitama. When utilizing the minimax algorithm, reasoning at higher depths should be preferred over utilizing theory of mind.

References

Aylett, R., Hall, L., Tazzyman, S., Endrass, B., Andre, E., Ritter, C., ... others (2014). Werewolves, Cheats, and Cultural Sensitivity. In *International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS)*.

Bard, N., Foerster, J. N., Chandar, S., Burch, N., Lanctot, M., Song, H. F., ... Bowling, M. (2020). The Hanabi Challenge: A New Frontier for AI Research. *Artificial Intelligence*, *280*, 103216. doi: https://doi.org/10.1016/j.artint.2019.103216

Byrne, R. W., & Whiten, A. (Eds.). (1988). *Machiavellian Intelligence: Social Expertise and the Evolution of Intellect in Monkeys, Apes, and Humans*. Oxford University Press.

De Weerd, H., Verbrugge, R., & Verheij, B. (2013). How much does it help to know what she knows you know? An agent-based simulation study. *Artificial Intelligence*, *199-200*, 67-92. doi: https://doi.org/10.1016/j.artint.2013.05.004

De Weerd, H., Verbrugge, R., & Verheij, B. (2014). Theory of mind in the mod game: An agent-based model of strategic reasoning. In *ECSI* (pp. 128–136).

Dupuis, N. K. (2022). *Theory of Mind for Multi-Agent Coordination in Hanabi* (Unpublished doctoral dissertation).

Lerer, A., Hu, H., Foerster, J., & Brown, N. (2020). Improving policies via search in cooperative partially observable games. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, pp. 7187–7194).

Premack, D., & Woodruff, G. (1978). Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, *1*(4), 515–526.

Rabinowitz, N., Perbet, F., Song, F., Zhang, C., Eslami, S. M. A., & Botvinick, M. (2018, 10–15 Jul). Machine theory of mind. In J. Dy & A. Krause (Eds.), *Proceedings of the 35th International Conference on Machine Learning* (Vol. 80, pp. 4218–4227). PMLR.

Russell, S. J., & Norvig, P. (2016). Alpha-Beta Pruning. In *Artificial Intelligence: A Modern Approach* (pp. 167–171). Pearson.

Weiner, T. (2018). *Onitama.* `https://github.com/TrippW/onitama`. GitHub.

A Appendix

	p-values and Z-statistics for all agent pairs	
Agent Pair	p-value	Z-statistic
Depth 3 vs. 3	0.823	-0.224
ToM_0 vs. ToM_0	0.395	-0.850
Depth 4 vs. 4	0.929	0.089
ToM_1 vs. ToM_1	0.227	1.208
Depth 5 vs. 5	0.447	-0.760
ToM_2 vs. ToM_2	0.655	0.447
Depth 4 vs. 3^*	$1.95 \cdot 10^{-98}$	21.058
ToM_1 vs. ToM_0^*	$6.65 \cdot 10^{-66}$	17.147
Depth 5 vs. 3^*	0.000	52.241
ToM_2 vs. ToM_0^*	$8.01 \cdot 10^{-56}$	15.740
Depth 5 vs. 4^*	$1.07 \cdot 10^{-303}$	37.250
ToM_2 vs. ToM_1^*	0.00091	3.318
ToM_1 vs. Depth 4	0.899	0.126
ToM_1 vs. Depth 5^*	0.000	-45.916
ToM_2 vs. Depth 4^*	0.00073	-5.819
ToM_2 vs. Depth 5	0.282	-1.075

Table A.1: The results of all the proportion tests performed in Section 3. Significant results are marked with a *, meaning the p-value is below the significance level of 0.05.