On the added value of Theory of Mind in Artificial Intelligence

A case of poker playing algorithms

Mick L. van het Nederend
10020799

Bachelor thesis
Credits: 18 EC

Bachelor Opleiding Kunstmatige Intelligentie

University of Amsterdam
Faculty of Science
Science Park 904
1098 XH Amsterdam

Supervisor
Dr. Nina Gierasimczuk

Institute for Language, Logic and Computation
Faculty of Science
University of Amsterdam
Science Park 107
1098 XG Amsterdam

June 26th, 2015
On the added value of Theory of Mind in Artificial Intelligence

A case of poker playing algorithms

Mick van het Nederend 10020799

Abstract

From a societal, psychological and evolutionary perspective, many benefits can be ascribed to Theory of Mind in humans. Therefore, it is likely that for artificial agents there also are a lot of gains to be had when using this trait. Currently, it has mainly acquired interest in AI when it comes to game-like scenarios, where the actual benefit of its usage is often left unmeasured. This thesis makes a case in favor of increasing the usage of Theory of Mind in AI, by measuring the increase in performance in different poker playing algorithms. These algorithms, three in total, all make use of a different order of weak Theory of Mind, enabling methods of quantification as means of evaluation. These different algorithms are then played against each other to determine the effective benefit of certain orders of Theory of Mind. The results show that especially for the first order Theory of Mind an increase of performance is noticable. This research thus endorses the claim that Theory of Mind could induce additions towards AI.
1 Introduction

Human beings are social animals that live and operate in groups wherein interaction among their members is ubiquitous. One of the ways of enhancing the performance in a social group is to be able to understand others, to be able empathise. This capability is a necessity, for without it one could be regarded as a "(...) 'hermit', and at worst diagnosed with a disorder like 'psychopathy' or 'autism'". (Rameson, Lieberman 2009) Furthermore, research suggests that the extent to which one can perceive others is an important factor in financial success. The capacity to understand the beliefs, emotions, desires and goals of someone other than oneself will be referred to here as Theory of Mind (ToM).

From a neurological perspective, Preston and De Waal (2002) have suggested that the usage of theory of mind in humans works as follows: the protagonist observes the situation of the antagonist and simulates that situation in their own head. Generally speaking, the reaction that is triggered is somewhat similar to the mental state of the protagonist. This model, the perception-action model, could be an explanation of the technical workings of what we have come to know as empathy. The model that is suggested here could expose difficulties for artificial intelligence (AI), for it needs to be seen whether or not a human-like notion of intentionality is a requirement for a human-like notion of empathy (and ToM): the difference between the simulated ToM reaction from an artificial agent should not differ too much from a simulated ToM reaction from a human in order to be accurate.

Currently, ToM is mostly used in the field of artificial intelligence in game-like environments. This could be explained by the fact that both the agent and its competitor (be it another agent or a human player) share similar goals, similar possible moves and a similar type of knowledge. In applications where this equipollency is less prevalent, the simulation as described in the perception-action model is more complicated. In game-like environments however, algorithms as straightforward as minimax can be said to use ToM, for it models the opponents best choice and takes into account the opponents situation and goals. In fact, every algorithm that takes into consideration the opponents possible moves and their likeliness uses ToM. One extreme example of this is described in De Weerd et al. (2013). This paper not only describes the benefits of ToM from one agent to another (first order ToM), but also from one agent to another and back - a knows that b knows that a knows (second order ToM) - and so forth. The paper describes a rock-paper-scissors environment and the added benefits of different orders of ToM. This is one of the few papers that both mention different orders of ToM and measure their added value.

In order to leap to a more real-world scenario that introduces many factors that do not solely revolve around the different orders of theory of mind, as in the end that is all rock-paper-scissors is about, the game of poker is discussed in this thesis. Apart from the usage of first order theory of mind (opponent modeling) and second order theory of mind (the modelling of how the opponent regards the agent), many other AI problems come into play in poker: imperfect knowledge, risk management, deception and unreliable information.
following I will briefly describe ToM from a psychological and evolutionary point of view as an entry point. Succeedingly the step towards AI is taken. For this thesis different poker algorithms are developed and will function as the common thread throughout this thesis to answer the question: To what extent do different orders of Theory of Mind contribute to the performance of game algorithms?
2 Theoretical foundation

The phrase Theory of Mind was first coined by Premack and Woodruff in their 1978 paper *Does the chimpanzee have a 'theory of mind'?*. These researchers investigated chimpanzees and suggested that they could be able to comprehend the mental states of others. They showed the chimpanzees video tapes of human actors experiencing simple problems, such as bananas hanging out of reach. Now, the primates were able to consistently point towards the photograph with the correct solution out of a set of photo's. This suggests that these animals are able to infer the goals, knowledge and beliefs of others. Being able to use ToM is not too obvious, however. Baron-Cohen et al. suggested for example, that autistic children do not have this capacity. Historian Lynn Hunt suggests that it was not until the 18th century, that due to the new French literature, which focused more on the psyche rather than the acts of the characters, humans started to develop empathy across classes and ethnicities. This would consequently be the foundation for the possibility of human rights.

Historian Lynn Hunt suggests that it was not until the 18th century, that due to the new French literature, which focused more on the psyche rather than the acts of the characters, humans started to develop empathy across classes and ethnicities. This would consequently be the foundation for the possibility of human rights. This implicitly suggests that a notion of theory of mind is always present, but it depends on the nurture to what extent this develops. This increment in the usage of ToM may partially explain how the violence per capita has been decreasing as suggested by Steven Pinker (2011).

About the benefit of ToM from an evolutionary point of view many theories have been suggested. Primates could function more efficiently in groups, by protecting each other and making sure everyone gets enough food. On the other hand, this improvement could reward lying and deceiving, as social structures enhances the level of competition among the species. This in turn would require detection mechanisms for these traits. Here the recursive ‘orders’ of ToM are already distillable. When speaking of ‘first order’ ToM, ToM from agent A about agent B is meant. The knowledge about someone’s knowledge about someone else in turn is named “second order” theory of mind. That someone else may also be initial person, like is the case in this thesis. Figure graphically displays this iteration of multiple orders. These orders can recursively be incremented.

2.1 Artificial Intelligence and ToM

The neurologists Preston and De Waal (2002) have suggested a model for the workings of Theory of Mind: the agent mimics the situation the antagonist is in, which generates similar mental states in the agents head. This perception-action model indicates that in order for ToM to work best, the protagonist should have a similar way of representing the world and a similar set of mental states as the antagonist. Not surprisingly, most AI implementations reason little about the beliefs and knowledge of the user, unless it is in a competitive environment, such as in a game. This is in agreement with the preceding: in competitive environments the goals, beliefs and knowledge of the agent and its opponent or team member share the same structure. AI tasks wherein the difference in ‘datastructure’ between agent and user (e.g. as is search engines) or subject (e.g. as in sentiment analysis) differs more, tend to be more difficult to
optimize. Of course, not in the last place due to the common sense problem and the difficulty of language, but also due to this difference in 'datastructure'. It makes formalizing and reasoning about the beliefs and knowledge simply more difficult.

As mentioned earlier, in games it generally is fairly simple: competitors often share the same goals, similar knowledge and similar sets of moves. Algorithms such as minimax model the best moves the opponent could do (so the worst move for the agent itself) and try to minimize the impact of such a move. The problem is that the exact contribution of multiple orders of Theory of Mind is fairly hard to quantify. De Weerd et al. (2013) proposes an interesting starting point for that: multiple rock-paper-scissors-like games were modelled. That research suggests that in that case the benefit of higher order theories of mind (higher than 2nd order) shows diminishing returns; it keeps getting better, but only slightly and only in particular situations. The games selected in De Weerd et al. however, purerly rely on theory of mind. In this thesis on the other hand, the focus is on a more complicated scenario, where theory of mind only is a part of the whole game. Three orders (zeroth, first and second) of theory of mind are modelled in order to play the game of poker.

2.2 Poker

Whilst poker algorithms are numerous, no real research has been done towards the contribution of the different orders of ToM. The reason why poker makes for an interesting game, is that there are multiple difficult AI problems: imperfect knowledge (the opponent’s cards and the cards to show up on the board are unknown), risk management (betting strategies and their consequences), agent modelling (identifying the opponent’s playstyle), deception (bluffing and varying playstyles). Neumann proved that in theory, for every zero-sum game (a game where the total gain of one or more players is equal to the loss of one or multiple other players), an equilibrium strategy is possible. An equilibrium strategy, meaning that on the long run, it would be the optimal tactic. In practice however, it is not so straightforward. These equilibria do not account for the playstyles of the opponent; it merely is the best way to play against someone else also using it. A better way is to determine the opponent’s playstyle and anticipate on its flaws. The algorithms developed have such a capacity in increasing quantities. Now, due to the timespan of the project and complexity it was not possible to implement pure theory of mind here: there is no reasoning from the opponents perspective. However, the outcomes of said reasoning are simulated. Therefore, this could be called ”weak theory of mind”. Combining this knowledge about poker and the points made by De Weerd et al., an interesting starting point is at hand. This leads us to the research question: To what extent do different orders of Theory of Mind contribute to the performance of game algorithms?

1For readers less familiar with the game, please find the rules of poker in appendix A, and a list with some of the terminology in appendix B
3 Research Method

In order to distinguish between different orders of Theory of Mind, three (albeit very similar) heads up Texas Hold’em-playing algorithms are created: one ‘baseline’ zeroth order Theory of Mind algorithm that does not take into account the playstyle of the opponent, one that does, and one that not only models the opponent but also takes into account the knowledge the opponent has of the agent. The baseline will be the fundament of the first order ToM algorithm, which in turn will be the core of the final, second order ToM algorithm. This is shown in figure 1.

![Figure 1: Graphic display of different orders of theory of mind.](image)

The first step thus is to create an algorithm that plays poker without taking into account the playstyle of the opponent - this algorithm will be referred to
as 'ToM-0'. The problem is however, that such a baseline algorithm without any opponent modelling is nearly impossible to make whilst maintaining a high performance. The imperfect information (in this context, the hole cards of the opponent) needs to be estimated one way or another to be able to reason about it. Filling in the holes where no information is available can not be done by simply taking into account every set of cards, or by generating a random set; the range of the cards are to be determined. This determination comes forth out of the playstyle of the opponent uptil now and the current move of the opponent. Therefore, a 'default' opponent model is constructed. This model will function as a generic, middle-of-the-road opponent model that completes this incomplete information in a simple yet effective fashion. In practice this comes down to the following: all three algorithms have certain values that model the opponent and the knowledge the opponent has of the agent, with the difference that these values are static in some algorithms, and variable in others. Therefore one could argue that the first algorithm is not completely devoid of theory of mind.

The first and second order Theory of Mind algorithms, ToM-1 and ToM-2 respectively, both use this default opponent model too, but only as a starting point. This makes sure the first few hands of the game do not create a ridiculous model due to the high variance. Every hand is remembered and added to the model, capturing the playstyle more and more along the way. In the ToM-2 algorithm, second order ToM is to be used. Generally in poker, this comes down to impede the modelling of yourself. Which is best done by being unpredictable: "(Unpredictability) makes it difficult for opponents to form an accurate model of your strategy. By varying playing strategy over time (...), opponents may be induced to make mistakes based on an incorrect model". (Billings et al., 1998, p. 232) The differences between these algorithms are graphically displayed in table 1.

<table>
<thead>
<tr>
<th></th>
<th>Keep track of hand history of the opponent</th>
<th>Use a varied gamestyle for means of unpredictability</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToM-0</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ToM-1</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ToM-2</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 1: The differences among the three algorithms

For this reason, it is fair to say that this is weak Theory of Mind: the practical outcomes of the different orders are simulated (the opponents playstyle is modeled and unpredictability is implemented), but the actual representation of the game the opponent has, is not recreated.

---

2Note that this greatly speaks in favor of the usage of Theory of Mind - without taking into account the goals and moves of the opponent it is nearly impossible to play a game even slightly well
3.1 Preflop

On to the implementation itself. Firstly, the hole cards are assigned a tier according to the table created by Sklansky and Malmuth. Although this model is developed for limit group games instead of no-limit heads up games, it is a good starting point, as the strength of starting hands generally does not differ greatly across game types. There are eight tiers, the strongest being tier one, the weakest being tier eight. The cards that do not fit into any tier will be considered tier nine here. The tiers are assigned as shown in table 2.

```
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>K</th>
<th>Q</th>
<th>J</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>K</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>Q</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>7</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Table 2: Sklansky and Malmuth hole cards ranking system. The sets on the top-right half of the table represent suited hands, whereas the sets on the bottom-left of the table stand for unsuited hands.

The algorithm plays fairly simple preflop. The higher the tier, the more it is willing to commit. Additionally, when the agent is on the dealer button, it is also willing to commit more than when it is big blind. The bets are fairly simple 2-bets, 2.5-bets and 3-bets. For more information, please consult the code. Its location and use is explained in appendix C.

3.2 Preflop Evaluation

The preflop moves and eventual pot commit are used to generate an approximate for the possible hand tiers of the opponent. This ‘aggressiveness’ factor is added to the descending list of previously added and default aggressivenesses of that current situation (e.g. ‘preflop, in position’). The position of this new added value in the list determines the approximate hand strength; if this position is $p$, the hole cards of the opponent are categorised as belonging to the best $\frac{p}{\text{length(aggri.list)}}$ hands. Every hole card tier has a certain occurency frequency.

---

4See appendix B - ‘x-bet’
These frequencies are compared with the value computed using the aggressiveness. The closest frequency will then determine the tier. This is shown in figure 2.

![Diagram](image)

**Aggressivenesses – preflop - in position**

1. Newly added value at $p = 4$
2. $4/20$ is closest to $240/1326$
3. Conclude: opponent’s range is T1-T5

**Cumulative occurrences**

- T1: 28/1326
- T2: 58/1326
- T3: 92/1326
- T4: 142/1326
- T5: 240/1326
- T6: 359/1326
- T7: 404/1326
- T8: 592/1326
- T9: 1

**Figure 2: Visualisation of the preflop evaluation**

### 3.3 Post flop

Once the tier of the opponent is approximated, every post-flop move of the villain is closely kept track off. The aggressiveness is again determined using this information. The next step is to determine both the win chance and the fold chance of the situation.

#### 3.3.1 Expected win chance

The win-chance is determined by Monte Carlo sampling. Every set of hole cards in the range (tiers) of the opponent is generated. For every one of these sets then random boards are sampled. At the flop, 1000 turns and rivers are sampled, at the turn, 200 rivers are sampled and at the river no board sampling is required as all the community cards are present. For every one of these boards it is determined whether or not the hero would have won from the opponent’s generated hole cards. This yields a list of booleans for every one of the hole cards, the percentage of which is true then is the sampled win chance for said hole cards. These percentages are in turn placed in an ascending list. Similarly to the preflop evaluation, the aggressiveness is placed into an aggressiveness list. The location of this value in the aggressiveness list tells us something about the hand strength of the opponent. If this location is at $p\%$ of the list, the average of the first $p\%$ elements in the win chance list is taken as the actual approximated win chance.
3.3.2 Fold chance

To determine the chance the opponent will fold, two factors are taken into account: the amount of times it has folded previously to a bet on the current street in the current position, and how committed the villain is to its current hand. The former is relatively easy, the latter is slightly more complicated. For this implementation we have chosen to work with a combination of the actions of this hand (check / call / raise) and the pot commitments on every street. These are compared with the actions and pot commitments overall. It is fairly similar to determining the quality of the hand of the opponent. The reason for taking into account the pot commitment is as follows: the more a player has committed to the pot, the less likely that player will fold later on, regardless of its hand strength. [12]

3.3.3 Combination

Bluffing without having any perspective on improving on the hand is risky and it tends to be better to semi-bluff: betting with a mediocre hand which could very well improve on the remaining community cards. [12] Therefore it is wishful to combine both the expected win chance and fold chance factors. The combined chance then is: \( P_{\text{win}} + (1 - P_{\text{win}}) \times P_{\text{fold}} \). This final probability, \( P_{\text{win or fold}} \) then determines the hero’s playstyle. Generally speaking, depending on position and preceding moves, a bet is made if this probability exceeds a certain threshold (0.4 in position, 0.6 out of position). This bet then is a value close to this probability times the size of the pot. Similarly, a call is made when \( P_{\text{win or fold}} \) within reasonable range (10%) to the percentage of the pot bet by the villain.

3.4 First order ToM

In the 0th order ToM algorithm, the aggressiveness values are not saved. They are merely inspected in terms of how their position in the aggressiveness lists would be. In the other algorithms, these values are saved. This enables the program to make a much more precise model of the opponent’s playstyle. In practical terms this means that the performance should not increase over time in the zeroth order ToM, whereas it should show improvement in the other algorithms.

3.5 Second order ToM

As stated before, in this implementation the ToM-2 algorithm only entails a small aspect of second order ToM. Instead of merely reasoning about the villains knowledge about the hero, it could be possible to reason about knowledge the other has about another agent. But since this is a one versus one environment, the latter is excluded from the algorithm. The most important aspect in this case is to implement a notion of unpredictability to the algorithm. In order to achieve this, a looseness factor is introduced. This double has a random varying value based upon a normal distribution with mean 1.00 and sigma 0.2. The
loosenessfactor is newly generated every hand. When the win chance and the fold probabilities are computed, these values are multiplied by this factor. This makes for a varying preflop range and postflop gamestyle, causes it to be more difficult to model reasonably.
4 Results

Due to the great variance that follows from the nature of poker, it is required to play a vast amount of games before any statements upon performance can reasonably be backed up. Due to the timespan of this thesis it was not pleasing to get enough games against human testers going to accurately measure the different algorithms. A different approach is to let the algorithms play against each other. This provides a way of getting many hands to be played in rapid succession. First, it is required to get a reasonable idea how big the variance in this context is. That is done by having ToM-0 play against another ToM-0 for $30 \times 100$ hands. The standard deviation is of the average cumulative gain for every 100 hands is around $2.0 \times 10^3$.

Hands are played in two possible oppositions: ToM-1 vs ToM-0 and ToM-2 vs ToM-1. The reason that ToM-2 is not tested against ToM-0, is that according to strong Theory of Mind, ToM-2 would foresee that it is not modelled by ToM-0. Therefore unpredictability is pointless and, and ToM-2 would act similar to ToM-1. There are six-hundred hands played ten times each. After each one of the ten iterations, the opponent model resets. The cumulative profit of the higher order algorithm then is smoothed, so that the average over one hundred (49 preceeding, 50 succeeding) hands is taken per datapoint. This is done to smooth the graph for both visibility purposes and variance exlusion. The results are shown below.

4.1 ToM-1 vs ToM-0

As shown in figure 3, the ToM-1 algorithm shows an improvement over time. In addition to the average over the measurements, the two measurements with the highest absolute value and the two measurements with the absolute average value are shown. The smoothed average cumulative gain starts out very slowly, presumably because the opponent model needs to be improved and then starts to rise increasingly fast. At around 500 hands this increase seems to be coming to a halt. At that time, the cumulative profit is around $8 \times 10^3$. The highest and lowest runs show that the lower bound has a much lower absolute value than the upper bound: -4982 vs 21160, respectively. The question now is, whether or not these results are significant.

Because the standard deviation of every set of hundred hands here is known, the average smoothed cumulative gains and their significances at every 100th hand are as follows:

- Hand 100: $5.8 \times 10^2(\sigma_{100} = 2000 \implies 0.9\sigma)$
- Hand 200: $8.1 \times 10^2(\sigma_{200} = 2000 \times \sqrt{2} = 2828 \implies 0.3\sigma)$
- Hand 300: $2.4 \times 10^3(\sigma_{300} = 2000 \times \sqrt{3} = 3464 \implies 0.3\sigma)$
- Hand 400: $6.4 \times 10^3(\sigma_{400} = 2000 \times \sqrt{4} = 4000 \implies 1.6\sigma)$
- Hand 500: $8.8 \times 10^3(\sigma_{500} = 2000 \times \sqrt{5} = 4472 \implies 2.0\sigma)$
- Hand 600: $8.6 \times 10^3(\sigma_{600} = 2000 \times \sqrt{6} = 4898 \implies 1.8\sigma)$
Since the final average result is $1.8\sigma$ away from the equilibrial state, it looks like the results are quite significant. Again, the stagnation around hand 500 is clearly visible. One could say that the algorithm has learned all it needs to know; extra information would only be duplicate information. The reason that it differs per game is that the amount of flops/turns/rivers played may vary, as many hands do not even reach the flop phase.

4.2 ToM-2 vs ToM-1

Figure 4 shows the results of the games played between ToM-2 and ToM-1. The amplitudes are remarkably smaller in this scenario. In addition, the ToM-2 algorithm even turns out slightly weaker than ToM-1. It looks like the implementation of unpredictability here has no effect. And if it does, it most likely is negative.

Below the average cumulative smoothed cumulative gains as well as their significances are stated.

Hand 100: $-1.2\times 10^3 (\sigma_{100} = 2000 \Rightarrow 0.6\sigma)$
Figure 4: Performance of algorithm ToM-2 when playing ToM-1 over 10 iterations

Hand 200: $-1.2 \times 10^3 \sigma_{200} = 2000 \times \sqrt{2} = 2828 \Rightarrow 0.4 \sigma$
Hand 300: $-7.5 \times 10^2 \sigma_{300} = 2000 \times \sqrt{3} = 3464 \Rightarrow 0.3 \sigma$
Hand 400: $-7.6 \times 10^2 \sigma_{400} = 2000 \times \sqrt{4} = 4000 \Rightarrow 0.2 \sigma$
Hand 500: $-1.4 \times 10^3 \sigma_{500} = 2000 \times \sqrt{5} = 4472 \Rightarrow 0.3 \sigma$
Hand 600: $2.7 \times 10^2 \sigma_{600} = 2000 \times \sqrt{6} = 4898 \Rightarrow 0.1 \sigma$

Since none of these results show sigma’s over 0.5, it is fair to conclude that the ToM-2 implementation does not contribute to the whole.
5 Evaluation

There are a lot of shortcuts taken in the algorithms developed. Firstly, not every player will simply invest to the pot solely based on the hand strength. Sometimes it might be better to 'trap' the opponent - playing a strong hand slowly, waiting for the other to make a bet in order to gain more value for the hand. In addition, some players might overplay some hands such as suited connectors (a case where the hole cards are both suited and consecutive in order e.g. $\spadesuit 7$, $\spadesuit 8$) whereas they do not play their albeit strong hands such as KQ very often. Secondly, the initial parameters chosen in terms of betting strategies differ from person to person. This disfavors the zeroth order algorithm and the performance of the other algorithms during the first few hands, until these default values become less significant. Then, for the sake of speed, not every possible board is simulated during the monte carlo sampling. Increasing this would slightly raise the accuracy of the win chance approximation. All of these flaws are not necessarily a huge problem however, as both of the agents have these limitations. In this isolated testing programme one could argue a slightly 'dumbed down' version of poker is played by both of the partakers.

Furthermore, it is difficult to determine the actual statistical significance of the results. The algorithm isn’t the fastest, so running thole array of tests tenfold was not plausible within the time span, although it would have helped a lot in terms of statistical analysis.
6 Conclusion

In this thesis three poker playing algorithms have been created to answer the research question: ‘To what extent do different orders of Theory of Mind contribute to the performance of game algorithms?’. These algorithms use different orders of weak ToM: 0, 1, or both 1 and 2. ‘Weak’, because there is no actual representation of the states from the antagonist’s perspective. Instead, the partial consequences have been modelled. The zeroth order ToM-0 algorithm does not remember the opponents previous moves, thus can not build an accurate model of the opponent. The first and second order (ToM-1 and ToM-2 respectively) do keep track of the preceeding actions and thus develop a sense of the opponents playstyle. Finally, ToM-2 implements a notion of unpredictability. The claim in De Weerd et al. (2013) [15] that lower orders ToM lead to a big increase in terms of performances whereas higher orders ToM do not contribute as much seems to be confirmed: ToM-1 shows a statistically significant improve over ToM-0, whereas ToM-2 does not show such an effect.
7 Discussion and future work

It now has been suggested that weak theory of mind may very well contribute to quite an extent this particular situation. An interesting new research could be the implementation of strong theory of mind. What would happen if instead of simply changing the looseness-factor in order to be less predictable, the actual scenario where the reasoning about "what does he know that I know about him?" is investigated. This could lead to more sophisticated ways of dealing with this knowledge. For instance, agent A knows that agent B knows that agent A has been bluffing quite a bit with bets sized 50% of the pot. Now agent A could opt for playing a strong hand like that as well, tricking agent B into thinking that agent A has a weak hand (first order ToM). Additionally, player B could foresee that using second order ToM.

Secondly, instead of playing in a heads up scenario, a table of three or more players could make for interesting cases. There is even more imperfect knowledge, and ToM can be used in a transitive way (A knows what B thinks that C thinks).

Furthermore, from a broader perspective, many more implementations of theory of mind could be interesting. Especially when it comes to fields outside games, such as natural language processing (NLP). Since one of the fundamental difficulties in NLP is the fact that language is a tool to exchange thoughts rather than a stand-alone tool, gaining more insights in these thoughts makes it easier to understand the meaning of certain expressions. [3] [8]
8 References


A Texas Hold’em rules (heads up)

Both players start with an equal amount worth of chips, 1500 in this case. The goal is to gain all the chips of the opponent. The player who succeeds in doing that wins. Consequently, the player who has run out of chips has lost.

Every round, or hand, the players take turns in being assigned the ‘dealer button’. This button determines who is to move first. In addition, both of the players get dealt two cards which each player can only see. This phase (preflop), is succeeded by phases where cards that are visible to both players are laid face-up upon the table. These succeeding phases are named ‘the flop’, where three cards are laid upon the table, ‘the turn’, where one card is added, and ‘the river’, where a single final card is added to these so-called ‘community cards’.

At the beginning of every hand the player with the dealer button automatically commits an amount determined by the ‘small blind size’, 10 in this case, of his chips to the pot. Similarly, the other player puts in chips equal to the ‘big blind size’ to the pot, which is 20 in this case. The player with the dealer button may choose to meet the higher pot-commitment of the opponent (to ‘call’), to pass this hand (to ‘fold’), or to add even more chips to the pot than would be required to call (to ‘raise’). These bettings carry on until at one point either one player has folded, or until both players have had at least one turn and their pot-commitments are equal. If a player has folded, the other player wins all the chips that both players have committed that hand, and a new hand is started. In the latter case, a new phase, or street, is entered, where in turn a bet round takes place. In every street from the flop onwards, time the player without the dealer button starts with making a move. Note that not raising when the pots are equal is seen as ‘checking’ and not as folding, thus allowing the player to continue playing the hand.

If all streets and their betting rounds have been played without anyone folding, the player with the best hand wins. The quality of every hand is determined by the best combination of five cards using the two cards in hand (hole cards) and the five cards on the table (community cards). The hands are ranked as follows (from worst to best):

**One Pair** - Two cards of the same rank (e.g. ♥A, ♥Q, ♦10, ♣10, ♠7)

**Two Pair** - Twice two cards of the same rank (e.g. ♦Q,♣Q,♦9,♥9,♣6)

**Three of a Kind** - Three cards of the same rank (e.g. ♠J,♣J,♦J,♣8,♦8)

**Straight** - Five cards with succeeding ranks, where an ace can be highest or lowest, but not wrap around (e.g. ♥J,♣10,♦9,♣8,♠7)

**Flush** - Five cards with equal suits (e.g. ♥A,♦Q,♠J,♦3,♣2)

**Full House** - Three cards of the same rank and two cards of the same rank (e.g. ♦A,♠A,♦8,♣8,♥8)
Four of a Kind - Four cards of the same rank (e.g. ♠K,♣K,♦K,♥K,♠7)

Straight Flush - Five cards with both equal suits and succeeding ranks (e.g. ♠K,♣Q,♥J,♠10,♠9)

Royal Flush - A special kind of straight flush with a leading ace (e.g. ♥A,♥K,♥Q,♥J,♥10)

A tie is only possible when the five ranks of the players’ cards are equal. If this is the case, both players receive half of the pot. When this is not the case, yet the players have the same denotation for their card combination, a tiebreaker is always possible. In the case that no denotation is given, or when both players have a straight, flush, straight flush or royal flush, the player with the highest card wins. If this card is equal, the second highest card is evaluated, et cetera. The remaining denotations first evaluate the ranks of the cards that determine said denotation, i.e. the two cards of the same rank in ‘one pair’, the four cards of the two ranks in ‘two pair’ (highest pair first), the three cards of the same rank in ‘three of a kind’ and the four cards in ‘four of a kind’. If these are equal, the remaining cards are again evaluated one by one from high to low. If both players have a Full House, the rank of which both players have three cards is evaluated before the rank of which both players have two cards. The ace is always the highest cards in this case, unless a straight (flush) is constructed from ace to five. In that case, the highest card would be a five.

B  Poker terminology

Bluf - Placing a bet while expecting the other player to have better cards.

Board - The common cards that everyone gets to see. Consists of the flop, the turn and the river. (Also see Community cards).

Community cards - The common cards that everyone gets to see. Consists of the flop, the turn and the river. (Also see Board.)

Flop - The first three community/board cards, all dealt at the same time.

Hand - (1) A 'round', consisting of the streets preflop, flop, turn and river. (2) The combination of a players hole cards and the board.

Heads up poker - Poker between two players.

Hero - The player from which point of view is looked at.

Hole cards - The two cards dealt to each player. These are the cards nobody else gets to see.

Limit Hold’em - A variant of poker where the bet sizes have an upper bound.
**No-limit Hold’em** - A variant of poker where the bet sizes do not have any upper bound.

**Pot** - The sum of the chips both players have invested.

**Rank** - The value of a card, i.e. 2,3,4,5,6,7,8,9,10,J,Q,K or A.

**Read** - Obtaining information about the hand strength of the opponent through its actions and the ways of performing those actions.

**River** - The fifth community/board card, deal separately.

**Stack** - The amount of chips a player has.

**Street** - The phase a round or hand is at. This could be either of the following: preflop, flop, turn, river.

**Suit** - The icon of a card, i.e. hearts (♥), diamonds (♦), spades (♠) or clubs (♣).

**Turn** - The fourth community/board card, dealt separately.

**Villain** - A player that is not the hero.

**X-bet (x being a number)** - A preflop raise being \(x\) times the big blind value.

## C  Code

The code can be found at [https://github.com/Mickmickmick/Poker](https://github.com/Mickmickmick/Poker)

It is a C# .NET / WPF application.

In order to alter the algorithms the `MainWindow.xaml.cs` file is to be consulted. To play against the computer yourself set the `VS_HUMAN` boolean to true. Furthermore, in the `InitializeGame` method the specifications about the algorithms are located. The `Villain = new AI(0, 1);` constructor call uses the first argument to specify the type of algorithm. For ToM-0, keep this value 0. For ToM-1 change it into a 1, and for ToM-2 a 2. Similarly the Hero’s algorithm can be chosen if the choice is made to let the algorithms play each other; if the `VS_HUMAN` boolean is false.

The respective orders are called from the `AI` class and called `OpponentModel_DEFAULT`, `OpponentModel_FIRST_ORDER` and `OpponentModel_SECOND_ORDER`. 