UTRECHT UNIVERSITY

Negotiating with Deceptive Agents in Mixed-Motive Interaction Environments using Theory of Mind

by

Paktwis Homayun

A thesis submitted in partial fulfillment for the
degree of Master of Science in Artificial Intelligence

in the

Science Faculty
Department of Information and Computing Sciences

August 2018

"The cruelest lies are often told in silence"

- R. L. Stevenson

UTRECHT UNIVERSITY

Abstract

Science Faculty
Department of Information and Computing Sciences

by Paktwis Homayun

Deception is the act of spreading a false belief through communication or actions, which is commonly used among people in situations where motives differ. Being able to perform such act is because of an ability called Theory of Mind. This is the ability to reason about the mental content of others, which often happens naturally in people. Higher orders Theory of Mind enable reasoning about how others use this ability. Such ability is used by people to get an idea of what someone's intentions are when that person performs a certain action. Using this principle a prediction of future actions can be made, which is a key aspect of deception. In competitive negotiation settings, it is expected that higher orders of Theory of Mind enable the agent to look further into the negotiation process that in turn can lead to a higher payoff through deception. This thesis explores the conditions needed for deceptive behaviour to emerge and which order Theory of Mind enables deceptive behaviour in computational agents using a mixed-motive interaction environment. An environment called Colored Trails is built to achieve this. The environment enables the agents to negotiate with each other using various proposals. Here, the agents have partial information about each others preferences which allows for the investigation of deceptive behaviour for different orders of Theory of Mind.

Acknowledgements

I would like to thank my project supervisor *Frank Dignum* for giving me the opportunity to go to Melbourne and work on this research.

Also my big gratitude goes to my secondary supervisors *Michael Kirley, Wally Smith and Liz Sonenberg* for their help and guidance throughout this project at Melbourne University.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Deception is usually seen as a human characteristic involving complex interactions and thought processes using different areas in the brain[3]. Some behaviour is deceptive if it makes someone perform an action that is better for yourself, given that other options were available that would have lead the receiver to choose a different action. Deceiving someone implies changing that persons beliefs, subsequently the decisions that will be made are based on those beliefs. However, having inaccurate beliefs does not always mean that deception has occurred. Different scenarios are possible where this holds, an example is where a persons' action does not depend on a certain inaccurate belief that has been induced by the deceiver. Ones beliefs may be inaccurate due to poor observation of the environment, wrong reasoning about the observation or some other factors that the deceiver is not responsible for.

When interacting with someone, people tend to reason about the mental content of that person which includes beliefs, desires and intentions. Reasoning about a persons behaviour and the above mentioned mental content is called Theory of Mind (ToM)[4]. Using Theory of Mind a person can make better predictions of how someone will react at a certain situation. Reasoning about a persons' mental content requires the so-called *first-order* Theory of Mind. An example could be: "Person A knows that person B wants to get some food", where person A is using first-order ToM. This technique is also used to reason about the Theory of Mind of others which enables a person to get an idea of how someone is interpreting the behaviour of others including himself, this is seen as higher-order ToM. An example of a second-order ToM reasoning is: "Person A knows that person B knows that person A is hungry". Unlike lying, if a person wants to deliberately deceive someone, he needs to have a mental model of how his behaviour is seen by the addressed party. Therefore Theory of Mind is necessary in order to achieve deception that is specifically aimed at that person [5][6]. Of course there exist other forms of deception, where a person is able to deceive someone without the need for

reasoning about that persons mental content, which is called *unintentional deception*. This form of deception will not be discussed here as it is outside the scope of this study.

1.1 Research Outline

There exist many different forms of deception where each conceal the truth in some way. This study focuses mainly on deception in the form of *lying by omission*[7] and *implicit deception*[8]. Such type of deception involves telling most of the truth, however leaving some important information out that change the perception of the situation completely, while only using actions. In order to investigate how such behaviour emerges in computational agents and what the conditions are that allow it to happen, an environment is needed that allows for cooperative as well as deceptive behaviours. For this purpose a simulated environment called Colored Trails is implemented in the form of a mixed-motive game[9] with incomplete information. By using such environment the agents have the freedom to find different strategies, including deceptive ones. However in order for deceptive behaviour to emerge the agents need to have incomplete information about some aspects of the environment. This enables the agents to make assumptions about those aspects and reason about how the opponent perceives them. This reasoning process can be done by inferring the other players' value-function, or in other words by modelling his behaviour using observation and reasoning about his mental processes, which is introduced here as Theory of Mind. The value function can be seen as function that tries to find the best solution using a scoring function as seen from opponents' point of view. However, this value-function of some other player depends strongly on the behaviour of yourself, which happens through your value-function. So inferring someones mental processes which are inferring yours leads to an infinite regress[10]. This is solved here by using the idea of *Bounded rationality*[11][12] to constrain the depth of reasoning in other player as modelled by the agent, which describes that when someone makes decisions, his rationality is limited to the cognitive limitations of his mind and will not exceed the scope of the decision problem.

By using Theory of Mind the agents can mislead their opponent by giving false information, or try to cooperate by revealing that information. Within this game the agent is able to strategically deceive an opponent by changing his beliefs about the environment or the game state. In order to achieve this, the agent needs a model that represents opponents beliefs, desires and intentions. The agent can exploit incomplete information that is available to the opponent using this model, by performing an action that results in opponent acting to the agents' advantage.

Research Question Under what conditions can deception emerge in computational agents using Theory of Mind? In this study a mixed-motive environment will be used, which enables the agents to choose from competitive or cooperative actions. Intuitively agents will have incomplete information as deception relies on some information being unavailable to the receiving party. Representing mixed-motive situations is done using Colored Trails (CT) framework, originally designed by Grosz and Kraus[13]. Colored Trails is a board game consisting of colored tiles that is played by two or more people or agents. Two agents are implemented in this research, a proposer that initiates the communication and a responder that reacts to his actions. The framework is explained more in-depth in 3.1.

1.2 Research Steps

First step towards answering the research question involves a *one-shot* negotiation game, where players can trade colored chips in order to advance towards the goal. Both players have the incentive to trade chips with each other because the initial distribution of chips of each player does not allow them to reach the goal. After the trade is completed the distribution of the chips is final. This approach relies on the deceiver having *complete information* about the game state as opposed to the responder that has *incomplete information*. Here, to achieve deceptive behaviour the focus lies in hidden information or obscuring of real information by the deceiver. Having access to and being able to change several aspects of the environment, prior to being observed by the other player, gives the deceiver an opportunity to come up with a deceptive solution that will make the other player form inaccurate beliefs about the game state. This is often seen in conjuration of magic tricks. Prior to starting a magic trick, the magician prepares the environment that will enable him or her to successfully deceive a spectator. In this one-shot game the deceiver needs to make a model of his opponent and predict his behaviour in order to successfully find deceptive moves. Simulation experiments are performed on cases where an agent has prior access to the environment and cases where both agents have the same amount of information. This is done to make a comparison between the performance of the agents using complete information as opposed to one having incomplete information. The results are then compared using two sample t-tests to show whether there is a significant difference between the scores of the agents. Further, significance testing using Pearson correlation[14] is performed to show whether the results have a significantly positive or a negative correlation.

The above mentioned approach is chosen to get an idea of how the environment is used by the deceiver in order to get a higher payoff and why having more information is important. However this approach is too trivial where the deceiving agent cannot lose

given a right environment, as will be shown later on. Subsequently this research is used as a stepping stone towards a game where both agents have incomplete information, which is the information about each others chip sets. Giving both players incomplete information about the game state gives both agents an opportunity to change the beliefs of the other player, and thereby deceiving him [5][15].

Having incomplete information may induce deceptive strategies that either maximize own given score or minimize the score of the opponent. Getting a better idea about the incomplete information is done by frequent communication between the agents during a single game. Making sure that the opponent gets the wrong idea about that information gives the deceiving agent an advantage, because having more information is crucial in mixed-motive interaction games as will be shown from the results of the first case study described above. For this purpose the second case study utilizes a *repeated-game* scenario, where the agents communicate at every turn. During such communication phase both players negotiate on a trade until they reach some sort of agreement. Through communication both agents are able to reveal or obscure some information that is hidden to the other player. In this scenario there is a proposing agent that initiates the trade at every turn, and the responding agent can either accept, decline or make a counter proposal. After the communication phase both agents are advanced one step towards the goal, if possible. Agents having different orders of Theory of Mind are able to reason about the other agents' mental states differently, and thus achieving different scores through either deception or cooperation. This behaviour is studied by performing simulation experiments using different orders of Theory of Mind agents. The significance of the differences in scores are then again compared using two sample t-tests. Further, by defining what deception is in this model it can be shown which orders of ToM agents are able to achieve this and what are the conditions needed for it to occur. Recognizing the occurrence of a deceptive behaviour is defined in Chapter 4.7 and 6.1.11.

1.3 Hypotheses about Deception

Because first approach uses a first-order ToM agent that has complete information about the opponent and environment, it is expected that he will create a model of the opponent that is identical to the actual one. This leads to deceptive moves that always result in the agent gaining a higher score than his opponent, given the right environment. Because a zero-order ToM agent cannot model opponents beliefs and intents, it will not be used as a proposer for this approach. In the second approach both agents have incomplete information about the game state, so both agents have to rely on their beliefs when proposing a trade. Because higher orders of Theory of Mind enable the agents to predict the behaviour of agents that are using lower orders of Theory of

Mind, the expected result is these agents obtaining a higher score through deception and cooperation, given the right distribution of chips. For equal orders of Theory of Mind agents the outcome is expected to be the same as if they were zero-order Theory of Mind agents. This is because both agents will try to make sure that the other models his chip set incorrectly. Once an agent discovers that his beliefs are incorrect he will be forced to rely his initial zero-order beliefs. Using those beliefs the agent will not consider opponents beliefs anymore and thus play as a zero-order ToM agent. The above-mentioned expectations are described formally in the hypotheses below:

Hypothesis H_1 Deceptive behaviour results in a overall higher score for that agent than the score he would obtain with cooperation.

Hypothesis H_2 First-order ToM agents with complete information will generally receive a much higher score than agents with incomplete information.

Hypothesis H_3 First-order ToM agents are able to deceive Zero-order ToM agents.

Hypothesis H_4 Equal orders ToM agents play as if they were Zero-order ToM agents.

Hypothesis H_5 Equal orders of ToM deceive each other roughly equally often.

Chapter 2

Related Work

Deceptive behaviour in computational agents and reasoning about reasoning are widely studied topics, especially in AI. A deceptive action can occur in different forms, because there exist many approaches to changing someone's belief. De Rosis et al.[16] study deceptive actions through direct communication. In this setting, the deceiver is trying to deceive the other agent by convincing him of the falsity of a certain fact. This model uses Bayesian networks[17] to represent beliefs and probabilities to other beliefs. Using this system, the probability of a belief being true can be decreased or increased by manipulating some other belief that is connected to it. In this model, the deceiver uses his beliefs to represent the beliefs of his communication partner. This method however does not consider the other agent having different beliefs, which is often the case.

GOLEM Another study by De Rosis, Castelfranchi and Falcone focuses on deception through actions in a blocks world environment called GOLEM [18]. Because the agents have to build structures using shared blocks, both agents have conflicting goals which can lead to deception. Planning is done using the information known about the other agent. The agents in GOLEM do not possess second order reasoning about reasoning of other agents, so they don't know how the other agent will react or interpret their action. This means that they only might accidentally deceive the other agent instead of deliberately. The goals, plans and beliefs of others are not accessible to an agent so in order to achieve deceptive behaviour, the agents need to form models of other agents mental states. Similar to GOLEM this work considers deception through actions, or non-verbal mechanisms.

PsychSim Pynadath & Marsella developed a simulation tool for modeling multi-agent interactions called PsychSim [19], where the agents make decisions using their own beliefs

together with the beliefs about other agent's beliefs. Occurring changes in the world and acquired knowledge about the game make the agents update their beliefs. This means that the agents can influence each others beliefs based on actions.

Level-n Theory In behavioral economics, reasoning about reasoning of others is modelled through different techniques such as cognitive hierarchies[20] or level-n theory[21]. These models represent an agents complexity as the maximum iterated reasoning steps that are available to the agent. Here a level-n agent models all other agents as exactly level-n-1 agents. This assumption cannot be made when considering repeated game settings, where the agents can change their levels of reasoning. To infer the mental states of others in such game scenarios the agents can make use of Theory of Mind, which is first formulated by Premack & Woodruff [4]. This ability helps understanding the behaviour of others and therefore helps predicting the following actions.

Dynamic iterated reasoning The agents described in the second case study use Theory of Mind similar to dynamic iterated reasoning models, where the agents adjust their level of reasoning based on the behaviour of others. These similar models include game theory of mind[10] and weighted attraction learning[22]. The latter uses *Belief-based models*, which allows the players to choose different strategies that have high expected payoffs based on the beliefs that are formed by prior observation of others. The agents in this study have a similar behaviour where expected payoffs are formed based on prior observations of others, however reasoning about others' beliefs, intents and desires play a big role in forming those expected payoffs.

Chapter 3

Background

This chapter covers some background knowledge and tools that are used for this study.

3.1 Colored Trails

A widely used framework that allows for complex multi-agent interactions that involves modeling and learning of decision making is called Colored Trails framework (CT)[23][24]. This test-bed is developed with the intent to investigate the decision making process in multiple computational agents and people. CT is a board game that can be played by people or agents which consists of colored squares. Every player has its own piece on the board and a goal he or she has to reach. Further each player has colored chips that can be used to traverse through the board. A chip of appropriate color needs to be handed in to make a move onto a neighbouring square that shares the color with the chip. When a player does not posses the right chips to reach the goal he or she can initiate a trade with other players. The other player can accept the trade or refuse. The score is then computed based on the players distance away from the goal and the amount of chips in his possession. The complexity of the game can be adjusted by increasing the amount of squares or colors, adjusting the scoring function, changing goal conditions or observability of the board. An example of the game can be seen in Figure 3.1, where both players are indicated by letters A and B on the board and their chip sets below. The numbering in each colored box shows the amount of chips a player has in his possession.
Due to unavailability of Colored Trails framework on http://www.eecs.harvard.edu/ai/ct the decision is made to create an own version of the framework that is suitable

for the purposes of this research using new tools and data structures. The mathematical model of this framework will be discussed in-depth in the next chapter.

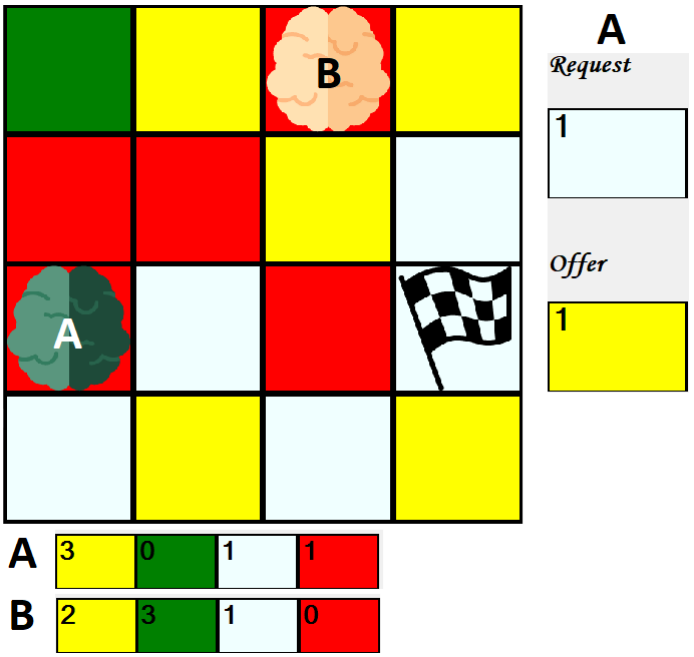


Figure 3.1: A two player CT game, where a player A has offered a yellow chip to trade for a white chip that is owned by the B player. The player A needs a white chip to reach the goal.

Initially for this study the game was implemented as a one-shot game, which means that after the trade is accepted or rejected the configuration of the chips is final and the agents are advanced towards the goal as far as possible and subsequently the score is computed. This setting comprises of one proposing agent that searches for optimal moves and corresponding proposals that will increase his and possibly the other agents' score and then proposes a trade. Further the game implements a responding agent that either accepts or declines the offer made by the other agent. This one-shot setting allows for investigation of agents' behaviour and deception when presented the proposing agent with complete information about the game and the other with incomplete information. Colored Trails framework is very useful when investigating agents interaction through non-verbal mechanisms, because the outcome of the agents depends strongly on each others' actions. However more interactions between the agents give a better chance to explore the workings of Theory of Mind and deception. In order to achieve this, the framework needs to implement a repeated-game. In this setting the agents interact with each other at every turn, while both having incomplete information about the environment. This incomplete information here is in the form of hidden chip sets, where both agents have initially no information about each others chips. During the interaction step the proposing agent can offer just one chip to exchange with one other, unlike the previous one-shot scenario. The responding agent can either accept, decline or make

a counter proposal. In case of the latter the roles of the agents are reversed and the previous proposer becomes a responder, having the same options to accept, decline or make a counter proposal. By interacting with each other in such way the agents reveal information about their chip sets to their trading partner, whether it is cooperatively or deceivingly depends on the environment, distribution of chips and the other agents' behaviour. This setting allows for *emergence* of deceptive behaviour instead of forced deceptive behaviour, as implemented in one-shot scenario. At the end of a turn the agents are advanced one square towards the goal along their shortest paths, given that they have the appropriate chips. If an agent lacks the right chips to advance to a neighbouring tile, he will stay at his position. Staying at the same position for five turns results in losing the game. In this scenario deception occurs when the deceiving agent successfully changes the beliefs of the other agent which results in that agent performing an action that is more advantageous for the deceiver. An agent can change opponents' beliefs in multiple ways, namely changing the beliefs about the chips in possession or changing the belief about what the intents, beliefs and desires of that agent are.

3.2 Theory of Mind

In order to achieve deception one must place himself in the opponents position and reason about how he would interpret your actions. This ability to reason about others' unobservable mental states such as beliefs, intentions, goals or desires is called Theory of Mind (ToM)[4]. Without such ability a person would only be able to reason about the behaviour of others, which is the observable content like someone performing an action. This is often seen as having zero-order ToM. Understanding that someone else has a different perspective is nothing new for humans, as this is a natural ability[4][25]. The use of higher-orders of ToM such as second-order allows a person to reason about how someone else is reasoning about someone. This recursive usage of Theory of Mind is believed to be a human ability, comparing to other animals. The emergence of social cognition in humans is explained by the Machiavellian intelligence hypothesis[25], which states that because of social cognition people can use deception and social manipulation to obtain an evolutionary advantage. Theory of Mind however, is still a theory because every human has only the ability to reason about the existence of its own mind and cannot inspect the mind of another person through introspection. The philosophical aspects, namely the Philosophy of Mind, will not be discussed here as it is outside the scope of this study.

In artificial intelligence, the ToM allows an agent to access or mimic another agents' mental state and reason about it[1]. There are different orders of ToMs, such as single-order ToM that can represent what another agent is thinking. However, second-order

ToM can not only represent the behaviour of single-order ToM but can also model what another agent thinks about someone else (including himself). The way a second-order ToM agent models the mental states of other agents can be seen in Figure 3.2. Here each model the agent A creates at first ToM level represents the beliefs of the other agents B and C. The models in the second ToM level represent the beliefs about the beliefs of agents B and C, creating a recursive tree structure.

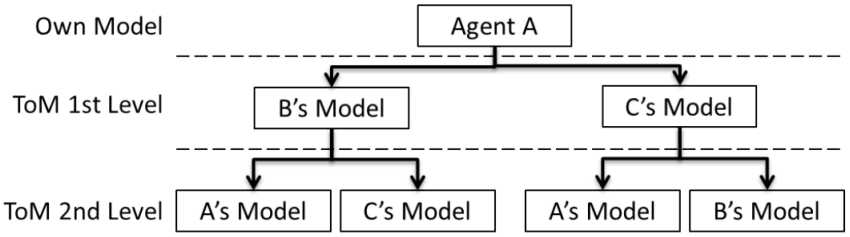


Figure 3.2: Multi-level hierarchy of other agents' models constructed by a second-order ToM agent. From *The Great Deceivers: Virtual Agents and Believable Lies*[1]

It has been shown that in most mixed-motive interaction games a higher-order ToM agent has a superior performance over its lower counterpart[26][27]. This is because agents using higher-orders of ToM are able to look further into the negotiation process and thereby make predictions of their opponents behaviour.

3.3 Path Planning

To navigate through some virtual environment such as a CT grid, an agent needs to have an ability of path planning. The most obvious path from the start position to the goal is the shortest path, that is available with the configuration of colours the agent owns. A path planning algorithm that considers this can be made using simple A-star algorithm[28]. Using this algorithm a set of possible shortest paths is computed, and for each path the chips are determined that are needed in order to take this path. The best path is chosen based on the beliefs, desires and intentions of the agents. Using his beliefs an agent is able to reason about whether the chips are obtainable from the opponent through trading, which he needs to take a certain path. If the agent believes that the opponent does not have the needed chip he will try to find other paths towards the goal. The general model of navigation on the board is given in chapter 4.5.

3.4 Deception Theory

Understanding the nature of deception helps with implementing and exploring deceptive behaviour in agents. The workings of deception or difference between perception-supported belief and the expectation based on memory is best described by studying the conjuration of magic tricks. Giving the opponent a false perception-supported belief in this project is inspired from conjuration of magic tricks[2], as the magician tries to hide some information or some real outcome from the observer. Simulating a process of a magic trick where some information is only available to the deceiving agent creates two parallel event sequences as seen in Figure 3.3. The method event sequence is the one that is understood only by the deceiving agent, which will be only revealed after the game ends. That gives the agent more knowledge than his opponent, which can be used to predict an action the opponent is going to make. While an another agent is not easily fooled through perceptual and attentional fallibility that is often used by magicians, there exist different approaches such as hidden information.
Handling with deception, whether perceived or actual, is explained by Interpersonal Deception Theory[6]. This theory focuses on the communication part between the sender and receiver as a whole rather than each individually. It explains the human behaviour when lying or when being lied to in a general way. IDT's model of deception is designed while keeping humans in mind however, the theory can be generalized for machine intelligence or agents.

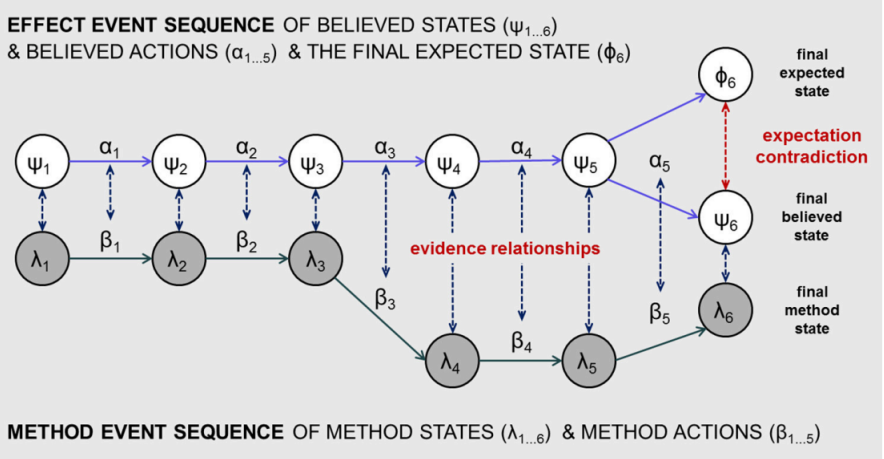


Figure 3.3: Two event sequences in a magic trick, effect event sequence is what is believed by the observer and method event sequence is what is believed by the magician (which is the actual truth) from *The Construction of Impossibility: Logic-Based Analysis of Conjuring Tricks*[2]

Chapter 4

Model

To explore how deception occurs using Theory of Mind, a specific test-bed is developed that resembles Colored Trails framework in terms of the environment and agents interactions. Colored Trails focuses on interactions between agents, where interactions are in the form of negotiation. Here the reasoning processes, performance and the choices the agents make can be analyzed and compared. The game is played by two agents on a 4 by 4 grid consisting of 4 colours, however the implementation allows for different game variants. For the purpose of simplicity this study focuses on the above mentioned configuration. Each coloured square on the board is randomly placed, which makes the amount of unique configurations of the board 4^{16}. On that board a random position for the goal is chosen. Each players goal is to get as close to the goal tile as possible by moving to a horizontally or vertically adjacent tile repeatedly. The initial positions for the players are selected randomly as long as they are not adjacent to the goal square. In both one-shot and repeated game variants the distances between both players and the goal are equal to make sure no player has a positional advantage. This study focuses on interactions between two players, and scenarios consisting of more players will not be discussed here.
In this study Colored Trails is defined using the following variables: A, $\mathbb{C}$, P, G, c_i^t, c_j^t, pos_i^t, pos_j^t and f. Table below shows the definitions of each variable:

4.1 Board

The board in the game is defined as a grid with a certain width and height that correspond to the amount of columns and rows respectively. Each square on the board, here named *a tile*, has its own colour, position and a set of neighbours N. The colours on the board are chosen from a known set of colours and placed randomly on each

Table 4.1: Colored Trails variables representation

A	Set of agents $\{i, j\}$ playing the game
$\mathbb{C}$	Set of possible chip set distributions
P	Set of possible positions on the board: rows $\times$ columns
G	Goal location on the board $G \in P$, same for both agents
c_i^t	Agents' i chip set at time t, where $c_i^t \in \mathbb{C}$
c_j^t	Agents' j chip set at time t, where $c_j^t \in \mathbb{C}$
pos_i^t	Agents' i position on the board at time t, where $pos_i^t \in P$
pos_j^t	Agents' j position on the board at time t, where $pos_j^t \in P$
f	Score function that takes an input s: $c^t \times pos^t \times G \rightarrow \mathbb{R}$, where $c^t \in \mathbb{C}$ and pos^t are agents' i or j chip set and position at time t and the function returns a real number $\mathbb{R}$

tile on the board. Because the first one-shot game approach of this study involves information that is unavailable to the responding agent, each tile has two properties namely *observed colour* and *actual colour*. During repeated-game scenarios the tiles only have one property which is the actual colour, because the amount of information each agent has is the same. After initialization process the proposing agent has access to the board before it will be observed by the other agent. In case of repeated-game scenario both observed colour and actual colour are the same for both agents, ensuring that no agent has more information available to him initially.

4.2 Chipsets

The agents negotiate trades of colored chips with each other, where the chips are distributed among the players in such way that it ensures that no player can reach the goal without trading. The shortest distance to the goal is not guaranteed, even after trading. When making a proposal, the agents may assign any value to the requested and offered chips, given that the game scenario allows it. During initialization, each colour is added randomly to the empty chip set a certain amount of times, with a minimal value of 0 and a predefined maximum value. This maximum value is $\frac{(width \cdot height)}{c}$, where c is the amount of colours on the board and width and height represent the amount of squares in x and y direction respectively. If a player is able to reach the goal with the resulting chip set, and thus does not require trading, a chip corresponding to a random tile along that path towards the goal is removed from the chip set. The last step of initialization is checking whether using both players' chip sets combined will lead both

of them to the goal. If that is not the case, then the chips that are lacking are added to the chip set. The resulting sets of chips allow for both agents to reach the goal after trading, which means that cooperation should be expected besides deception.

4.3 Players

During the initialization players are assigned random positions on the board, given that it is at least one square away from the goal and have the same distance to the goal. The agents can make observations of the board at any time, which allows them to see the goal position and the observed colours on each tile. Initially, each player is assigned a role of a proposer or responder where a proposer always initiates the trading process and responder reacts to his proposal. Both agents can observe their own chip sets and form beliefs about the chips needed to reach the goal. This is done by calculating the paths towards the goal and the chips needed to take that path. The chips that an agent is lacking are the ones he needs to trade with his trading partner, given that he believes that those chips are owned by the other agent. Those beliefs are formed based on his observation of the game. In one-shot case study the agent is able to observe opponents' chip set directly, whereas in repeated game the beliefs are formed through repeated communication and observation of opponents responses or requests. Further, the agents can represent their opponents beliefs and behaviour using Theory of Mind. Depending on the order of ToM, an agent can make a model of his opponent. This is done by observing the position of an agent on the board and his chip set. This chip set can either be known in case of a complete information game or approximated if the scenario uses incomplete information. Because the goal location is the same for both players, a non-zero-order ToM agent can form beliefs about opponents intentions given the above mentioned information.

At any time, an agent can compute his score using knowledge about his chip set, his position and the goal position. The scoring function used for this is explained in the next section.

4.4 Scoring Function

Both agents use the same function that assigns scores to them. The outcome of the function depends on a constant initial base score of the player, distance from the player to the goal, the goal score and the amount of chips that are in possession. This function is defined for an agent i as follows:

$$f^i(s) = B^i_{score} + g^i(s) + \sum_{c \in Chipset} (c_{weight} \cdot Col_{weight}), \tag{4.1}$$

where B^i_{score} is the initial score assigned to the player, c_{weight} and Col_{weight} are chip weights and colour weights respectively. The chip weight is a constant that is assigned during the initialization. Colour weights are variable and get their values based on the occurrence of that colour on the board. Lower amounts of colours on the board have a higher weight. The weights enable the agents to try to gather as many chips as possible, while trading the chips that have a lower weight. This ensures that the agents would not trade all their chips, but rather look for more optimal solutions. For repeated-game scenario both chip weights and colour weights are initialized as 0, which means that the scoring is computed only based on the initial score and the distance from the goal. This is done because in a repeated game the agents advance one step towards the goal at each time, handing in an appropriate chip. If such chip would have a weight then the agent would have no incentive to hand in his chip, as it will lower his score while he is not at the goal. Computing the score based on the initial score and the distance from the goal is denoted in the above equation as function $g^i(s)$, which is defined as follows:

$$g^i(s) = \begin{cases} G_{weight} & if \quad Manhattan(pos^i, pos^{goal}) = 0 \\ D_{weight} \cdot Manhattan(pos^i, pos^{goal}) & otherwise \end{cases} \tag{4.2}$$

Here, the G_{weight} represents the constant goal weight and D_{weight} is a *negative constant* distance weight. This distance weight ensures that a player would receive a higher score if he would advance one step towards the goal and a lower score if he would move away from the goal, as the *Manhattan distance* would decrease or increase. The distance towards the goal is computed by Manhattan distance, which only allows for movements in horizontal and vertical direction, excluding the diagonal moves as shown in Equation 4.3. If that distance equals to 0 then the player is at the goal and will thereby receive the goal weight. If the player is at least 1 step away from the goal then the function $g^i(s)$ will result in the amount of steps away from the goal multiplied by the negative distance weight.

$$Manhattan(pos^i, pos^{goal}) = \left\| pos^i_x - pos^{goal}_x \right\| + \left\| pos^i_y - pos^{goal}_y \right\| \tag{4.3}$$

4.5 Moving on the Board

The agents navigate along the board using their chip sets and the scoring function. If the colour of a neighbouring square is present in the chip set and advancing on it yields the highest score, the agent will move onto it. If moving on a square does not increase the score, the agent can still move onto it if that results in increase in score for movements that will follow after. Because both case studies are performed in different game environments, namely repeated and one-shot, movement on the board differs from step by step to immediate route towards the goal. However, the core functionality of the path planning method in both environments is not dissimilar. Making a decision to advance towards the goal onto a neighbouring square, given a players' current chip set, is shown in the equation below.

$$\max_{n \in N} f(n) > f(pos) \quad if \quad n_{color} \in Chipset \tag{4.4}$$

Here, N is the set of neighbouring squares and *pos* represents the current position of the player. For each neighbouring tile that result in same score, the agents repeat the process around their neighbours. An example is shown in figure 4.1, where agent i has four options to move onto the neighbouring tiles. Moving on tile number 2 is not possible given agents' i chip set, because he lacks an orange chip. Moving on tile number 1 gives the highest score among the possible movements, however other possible routes towards the goal need to be considered as well which include tiles 3 and 4.

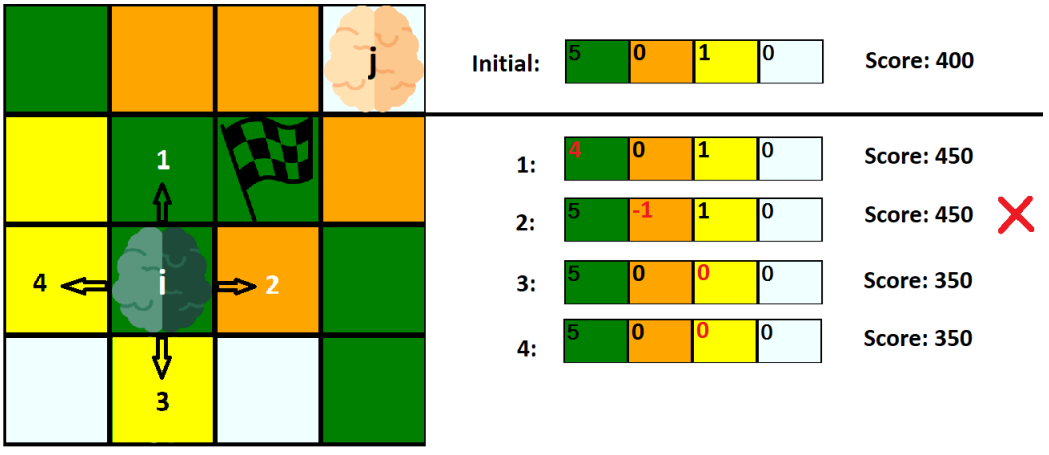


Figure 4.1: An example of possible movements available to agent i, given his initial chip set. The resulting chip sets are shown below the initial chip set, for each movement respectively. Moving onto tile 2 is not allowed because the player lacks the required chip.

4.6 Interaction between Agents

Depending on his role, an agent has to either make a proposal or respond to the given proposal. In both cases the agents make decisions about which action is best to perform. This decision making process relies on agents beliefs and the information available to him. This information includes other players' chip set and colours on the game board. Subsections below describe a basic decision making mechanism when performing the stated actions.

4.6.1 Making a proposal

When making a proposal, player i considers his model of the opponent, denoted as j'. The model is represented as a new player that has the same position on the board and the same chip set as agent j. This chip set can either be known, if player i has complete information about the game, or approximated. The latter case applies to an incomplete information game, where both agents don't know about each others chip sets. Using this knowledge about opponents chip set and the given board, player i can make a prediction of both scores he and the opponent can obtain given that his proposal is accepted. This predicted value is denoted here as *Expected Value* (EV). Using the response function of the modelled player j, player i can predict whether a proposal will be accepted or not. The precision of the prediction depends on how well player j is modelled by player i, which in term relies on precision of the approximated chip set. Given the current position, a prediction of the score for agent i at time t is computed as follows:

$$EV_i(p, c_i^t, c_{j'}^t) = \begin{cases} f^i(c_i^{t+1}) & if \quad Response_{j'}(p, c_i^t, c_{j'}^t) = accept, \\ f^i(c_i^t) & otherwise, \end{cases} \tag{4.5}$$

where p is some proposal, c_i and $c_{j'} \in \mathbb{C}$ are the chip sets of player i and j' respectively. Here $\mathbb{C}$ denotes all possible chip sets. The score for the given new chip set is computed if player j' accepts the offer, otherwise the current score is taken.

4.6.2 Responding to a proposal

Upon receiving a proposal, player j observes his own chip set and the board. If accepting the proposal will result in the best possible score while being higher than current score, and the requested chips are present in his chip set, player j will accept the

proposal. Finding the best proposal is done by looking at the opponents' approximated chip set. If this chip set allows for a better proposal, player j will decline the current proposal. The general response function at time t is defined as follows:

$$Response_j(p, c_i^t, c_j^t) = \begin{cases} accept & if \quad f^j(c_j^{t+1}) > f^j(c_j^t) \quad and \quad f^j(c_j^{t+1}) > f^j(c_{max}), \\ decline & otherwise, \end{cases} \tag{4.6}$$

where c_{max} is a chip set containing maximum chips given the game state. This variable is obtained by combining both chip sets of agents i' and j. Using this combination of chip sets the agent can find the best possible score. The scores are computed given the current position of the player. Note that i' is modelled by player j.

4.7 Finding Deceptive Moves

Changing the beliefs of a player happens through different actions. If that player chooses to take an action that yields him a lower outcome than some other action that was available to him before the change in his beliefs, the player can be considered as deceived. This is only true if the player would have chosen the better action if his beliefs were not changed. Thus in order to find out when deception occurs, some agents' beliefs need to be inspected before and after a change occurs. However, just lowering a players' beliefs does not necessarily mean that deception has occurred. If that player happens to have incorrect beliefs about the game state, correcting them by decreasing his options would not be considered a deceptive move. Similarly if that player would have chosen a lesser action in the first place, regardless of the change in his beliefs, then no deception has occurred. This means that besides looking at a players' beliefs, in order to find deceptive moves one also needs to know the intentions of that player as well. The models for recognizing deception in one-shot and repeated games are shown in chapters 5 and 6.

4.8 Finding Cooperative Moves

Because this is a mixed-motive setting, cooperation besides competition between the agents is expected. This is because the agents adjust their strategies based on the environment and the behaviour of the opponent. Recognizing cooperation is done in a similar way as recognizing deception. Having better options to chose from than initially

known can be the result of a cooperative action. An action is cooperative if it results in both players gaining a higher outcome. This happens when a player changes his trading partners beliefs in such way that it benefits both players. By cooperating with each other through bargaining the game essentially becomes a multi-objective maximization problem, with variable objective weights depending on the orders of ToM and the adaptability of the agents. However some situations may arise where cooperation can be followed by deception. This happens when the deceiver first wants to increase the overall maximum scores and subsequently take the largest piece for himself. The models for recognizing cooperation in one-shot and repeated games are shown in chapters 5 and 6.

Chapter 5

Case Study 1: One-Shot Games

This chapter describes the approach used for investigation of Hypothesis 1 and 2, namely whether deceptive behaviour results in a overall higher score than cooperative behaviour and whether first-order ToM agents with complete information in general receive a higher score than agents with incomplete information. Investigating these hypotheses gives further insights into deception and why an agent needs some hidden information in order for deception to occur. Additionally, the goal of this case study is to show how the agents communicate with each other through non-verbal mechanisms. The environment used for this is configured as a one-shot game. Reason for choosing a one-shot game scenario is to restrict the roles of both agents and their communication steps, which allows for a closer investigation of their individual behaviours. Such game scenario consists of one communication phase, where agent A makes a proposal to trade certain chips and player B either accepts the proposal or declines it. If the proposal is accepted then the resulting distribution of chips becomes final. The players then approach the goal as closely as possible by handing in chips of appropriate colour that corresponds to a tile the player steps on. The final score is computed once the players cannot advance any further. The next sections describe the model and simulation results for one-shot approach. Further, this approach is a stepping stone towards the next case study described in Chapter 6.

5.1 Model

This section describes a specific model for this case study, expanding on the general model introduced in the previous chapter. The agent i that takes the role of a deceiver initially has access to the game board. Using information about position and the chip set of his opponent agent i can decide to change the appearance of certain tiles in such

way that it benefits him. Because the configuration of the game allows for cooperative behaviour, allowing both agents to reach the goal, agent i can decide not to change the tiles. However, in some cases changing the tiles results in a higher score for agent i because he would give away less chips during the trading process. Subsequently the information about the board that becomes available to the other agent then might be *false complete*, which means that the information may appear as complete to that agent while in reality it is not, giving him a false perception-supported belief. This splits the game in two event sequences as described in chapter 3.4, namely effect event sequence that is believed by agent j and method event sequence that is known to deceiving agent i. Using their complete information about the environment, which includes own and opponents' chip set, the goal location and the colour of the tiles, both agents can find shortest paths towards the goal.

5.1.1 Navigation in One-Shot Games

Before finding the shortest paths towards the goal, the chip sets of both agents are combined. For each neighbouring tile from the initial position of the player is checked whether the combined chip sets allow for movement onto that tile. If it does then the tile is added to the path. This sub-path is then inserted into a *priority queue*[29] data structure. Priority queue allows for finding the minimum or maximum value in *O(1)* constant time and insertions and deletions in *O(log n)* time. This is very useful because insertions and deletions happen often. The minimum and maximum values of a path are determined based on the chip weights that build up that path. The path that has a minimum value in the priority queue is taken first and the process is repeated around the last position on that path. If a position that is added to the path equals the goal position then the path is removed from the queue and inserted into a set of possible paths. Similarly, if each neighbouring position around a tile is not reachable because the agent lacks the required chips then the path is removed from the queue as it does not lead to the goal. One step of this process is shown in figure 4.1, where the possibly reachable neighbours are added to the partial path which in turn is inserted into the priority queue. Using the priority queue data structure provides a sorted set of paths, starting with a path with the least cost. The algorithm terminates when the specified amount of paths is found. Then, for each path the needed chip set can be computed. With this chip set the goal location can be reached. The algorithm for finding optimal paths is shown in the Appendix section B. This algorithm essentially uses a repeated form of the equation given in 4.4, where partial paths are ordered in the priority queue based on the score they yield.

5.1.2 Agents

In this case study the agents have different amount of information available to them to show the importance of having complete information about the environment as opposed to incomplete information. The deceiving agent i has the role of a proposer, starting the trade process in every game. Further, he has an ability to change the appearance of the tiles on the board before the actual game is started. The changed colour on a tile becomes the observed colour, whereas the actual colour is hidden from the other agent j. Having access to the actual colour information gives the agent i an advantage over the other player. However in order to deceive the opponent, agent i needs to change the tiles and make a proposition in such way that it does not become suspicious to agent j. Proposition and response functions are explained in-depth in section 5.1.4.

Before proceeding to change the tiles agent i makes a model of his opponent, agent j, based on his observation. Because agent i has information about the other players' chip set and position on the board, he can make a perfect representation of his opponent. Finding the right tiles to change is done by first computing the best paths towards the goal. These paths are the shortest paths to the goal given that a player has both chip sets as shown in the previous subsection. Agent i then determines which chips he lacks in order to take one of the best paths. The chips that are not needed for that path can be used for changing the colours of tiles, here denoted as $C*$. First the neighbouring tiles N around the goal are taken, and ordered by the Manhattan distance to the goal pos^{goal}:

$$\forall n \in N_{goal}, \quad N_{goal'} = OrderBy(\min_{n \in N_{goal}} \left\| Manhattan(pos_x^n, pos_x^{goal}) \right\|) \tag{5.1}$$

Subsequently, for each of the given tiles a recursive Depth First Search[30] around its neighbours is used to check which tile can be changed into a certain colour. By looking at the response function of the modelled agent j', agent i can determine whether his proposal will be accepted for a newly changed board:

$$ChangeColour(N) = \begin{cases} n_{colour} = cl* & if Response_{j'}(p, c_i, c_{j'}) = true \\ ChangeColour(N_{n'}) & otherwise, \end{cases} \tag{5.2}$$

where $cl* \in C*$, p is the proposal that requests the colours needed to reach the goal and offers the colours that are not needed to player i. If the modelled player

j' declines the offer then the recursion starts around the ordered neighbours of the last position. This method is an approximation of Breadth First Search, which can't be used in recursive functions. It results in tiles being checked first around the goal. Using the response function of the modelled opponent, player i can find a layout of the board and the corresponding trading chips that will yield him the best path towards the goal, thus maximizing his score. However once a right tile is found the agent needs to backtrack through his previous changes because some of these changes are unnecessary for obtaining the chips needed from the opponent. This is seen in figure 5.1, where figure 5.1 A shows the state directly after a tile is found that will be used to force the opponent into making a wrong decision and 5.1 B is the state after backtracking. The green line in A shows the best path for player j.

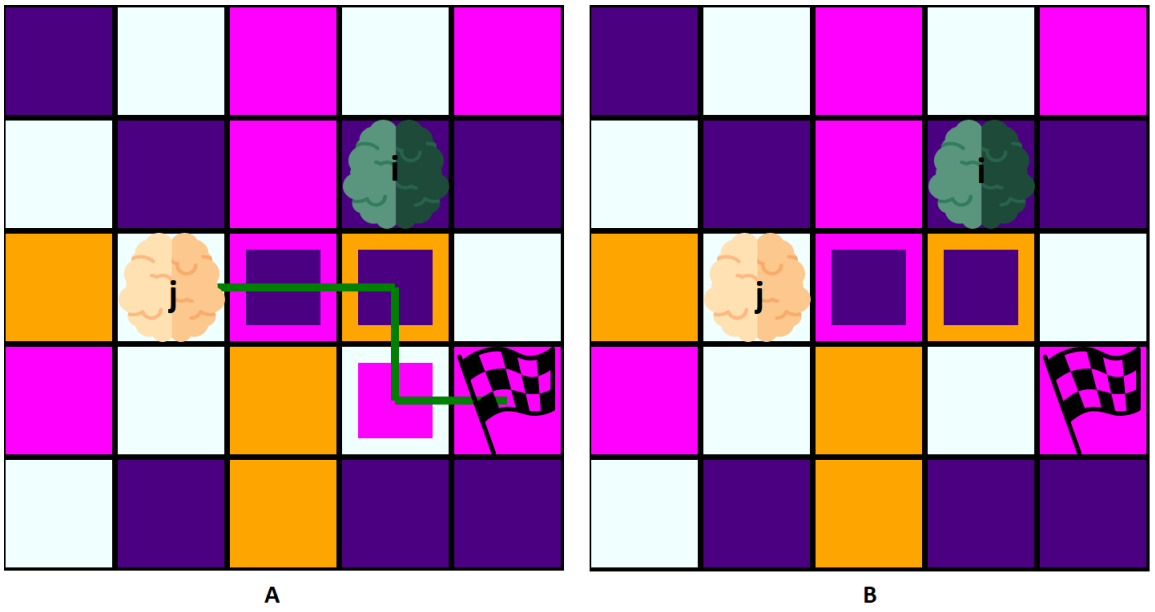


Figure 5.1: Changed tiles on the board, small coloured tiles are the observed colours and the ones behind is the actual colour. Figure A shows the conditions at the time when the last tile is changed that will yield a desired outcome for the deceiving agent i. Figure B shows the state after backtracking, where the pink observed colour is removed. This shows that the tile is not relevant for the desired outcome to happen.

5.1.3 Forcing deceptive moves

While changing the tiles, agent i tries to find which additional chips he should request in order to attain a higher score while making sure that agent j is not able to reach the goal. The paths towards the goal computed by agent j are based on his beliefs about the tile colours on the board. Because agent j can only observe the apparent colour of the tiles instead of the real colour, his predicted paths can be different from the actual paths towards the goal. Keeping that in mind agent i can find a proposal that agent j is willing to accept. This is done by first computing the opponents shortest paths

towards the goal, given that he has information about the real colours on the board. Subsequently agent i adds one chip needed to take the paths to the set of chips that he will request, if agent j has such chip in his set. Upon revealing the actual colours of the tiles after the trade is finished, agent j might not be able to reach the goal. This shows that agent i has a preference of deception over cooperation. However, if a game scenario does not allow for deceptive behaviour due to an unlucky distribution of colours on the board and chip sets, then cooperation is chosen instead. In such case the agent will try to find a proposal that increases both agents' scores.

5.1.4 Interaction Between Agents

In this game scenario agent i is assigned the role of proposer and j responder. Because both agents have complete information about each others preferences through visible chip sets only one communication step is needed. This step consists of player i making a proposal based on the information he obtained previously when changing the colours on the board. Responding agent j accepts the offer that strictly increases his own score, unless the agent finds that the offer does not coincide with his beliefs about the given board configuration.

Making a proposal

Agent i requests the chips he needs in order to take one of the best paths that resulted through changing the tiles on the board. Similarly if there are no tiles changed the agent looks trough best paths that he can take towards the goal. In both cases agent i considers opponents best possible paths. Both sets of best paths are here denoted as B_i and $B_{j'}$ for agents i and j' respectively. The agent i then considers the chip set needed to take the first best path $B_i \rightarrow c_i^{best}$ and the chip set his opponent needs for his best path $B_{j'} \rightarrow c_{j'}^{best}$. Because best path set contains ordered elements, the first element contains the most desired path. Requesting the desired chips is done as follows:

$$Request(chip) \quad if \quad chip \notin c_i \quad AND \quad chip \in c_{j'}, \quad \forall chip \in c_i^{best}, \forall chip \notin c_{j'}^{best} \tag{5.3}$$

Agent i requests the chip given that he does not have it in his chip set and according to that agents belief agent j does. Further, that chip has to occur in the chip set that agent i needs to take the best path while not being in agents' j best path chip set. Offering chips to player j is done in a similar way, namely:

$$Offer(chip) \quad if \quad chip \in c_i \quad AND \quad chip \notin c_{j'}, \quad \forall chip \in c_{j'}^{best}, \forall chip \notin c_i^{best} \tag{5.4}$$

Responding to a proposal

Upon receiving a proposal agent j considers the possibility that his opponent is trying to deceive him. In order to make a decision of accepting the offer, agent j verifies the offer and rejects it if one of the following rules does not hold:

- **Offered chips cannot be empty**, otherwise there is nothing for agent j to gain from the trade.
- **Requested chips need to occur in agents' j chip set**, the agent namely cannot make a trade with chips he does not have.
- **Opponent can not reach the goal without trading**, otherwise this would contradict the game mechanics where both agents are not able to reach the goal without trading.
- **Agent j can not reach the goal without trading**, contradiction of game mechanics, same as previous item
- **Requested chips need to form at least one path towards the goal**, out of all paths that become available to the opponent given that the trade is accepted, at least one need to contain all of the requested colours.
- **Agent j gets a higher score after the trade is accepted**, agent j cannot accept proposals that decrease his score.

5.2 Examples

This section shows two examples of one-shot games, where the proposing agent i initially changes the tiles. The colours on the board and the players' chip sets are randomly chosen, while restraining the condition that both players cannot reach the goal by themselves.

Example 1 Consider the board layout shown in Figure 5.2 A, where agent i is the proposer and j the responder. Both players' respective chip sets are shown below. Here the agents have different ways of reaching the goal by cooperating with each other, as seen from their chips. However before a cooperative move is considered, agent i first

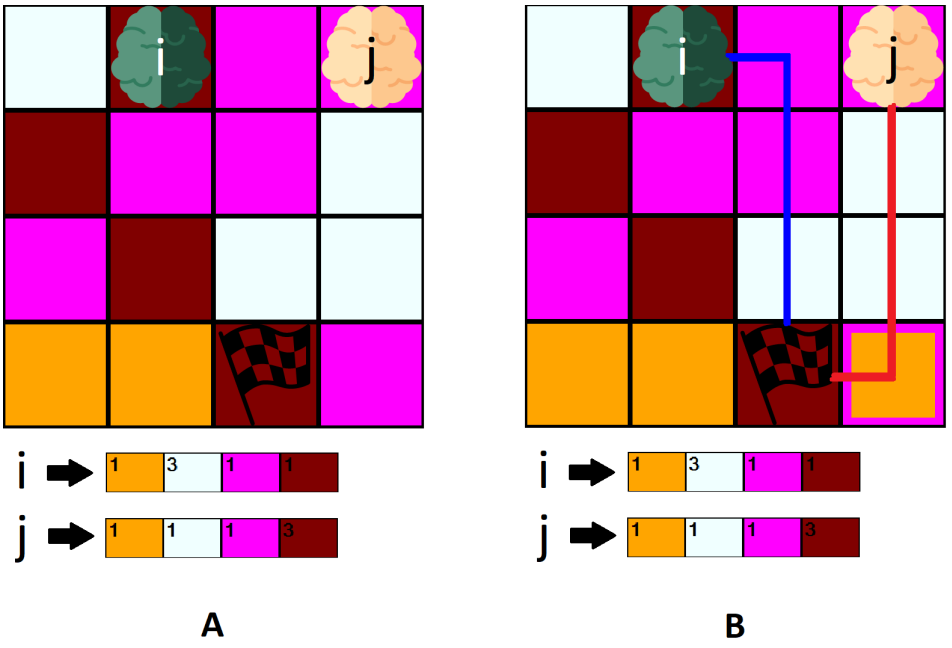


Figure 5.2: Two states of the game where A is the initial board with both players' initial distributions of chips and B is the state of the game after agent i changed the tile. Figure B also shows the paths towards the goal as computed by both agents.

looks for solutions that will only result in himself reaching the goal. In order for the proposal being accepted by the opponent, agent i needs to change the tiles on the board in a strategic way. The chips that are not needed to take his best path are used to achieve this. The agent recursively changes the tiles around the goal. After changing a tile, agent i tries to find a proposal that agent j is willing to accept. This is done by making proposals to the modelled agent j, which is constructed by agent i using the information available to him. Once the proposal is accepted and the real colours on the board are revealed the responding agent should be unable to reach the goal. If a board configuration does not allow for such behaviour then agent i makes a cooperative proposal, which allows for both agents to reach the goal. Figure 5.2 B shows the state of the game after agent i found a tile to change, where the red line indicates the best path for agent j and blue line for agent i. After the tiles are changed on the board, agent i makes a proposal to trade chips with his trading partner.

Figure 5.3 A shows the proposal made by agent i, requesting the colour pink that is needed to take his best path towards the goal. Note that in this case the requested colour is also the colour the agent concealed. By offering only one white chip the agent makes sure that agent j cannot reach the goal after the real colours on the board are shown. If agent i would have offered two white chips in return, agent j would accept that offer too. However then the agent would be able to take the path that goes through the three white squares and thus reaching the goal. Similarly agent i could also reach the goal by requesting one brown chip and offering a white in return. However in doing so

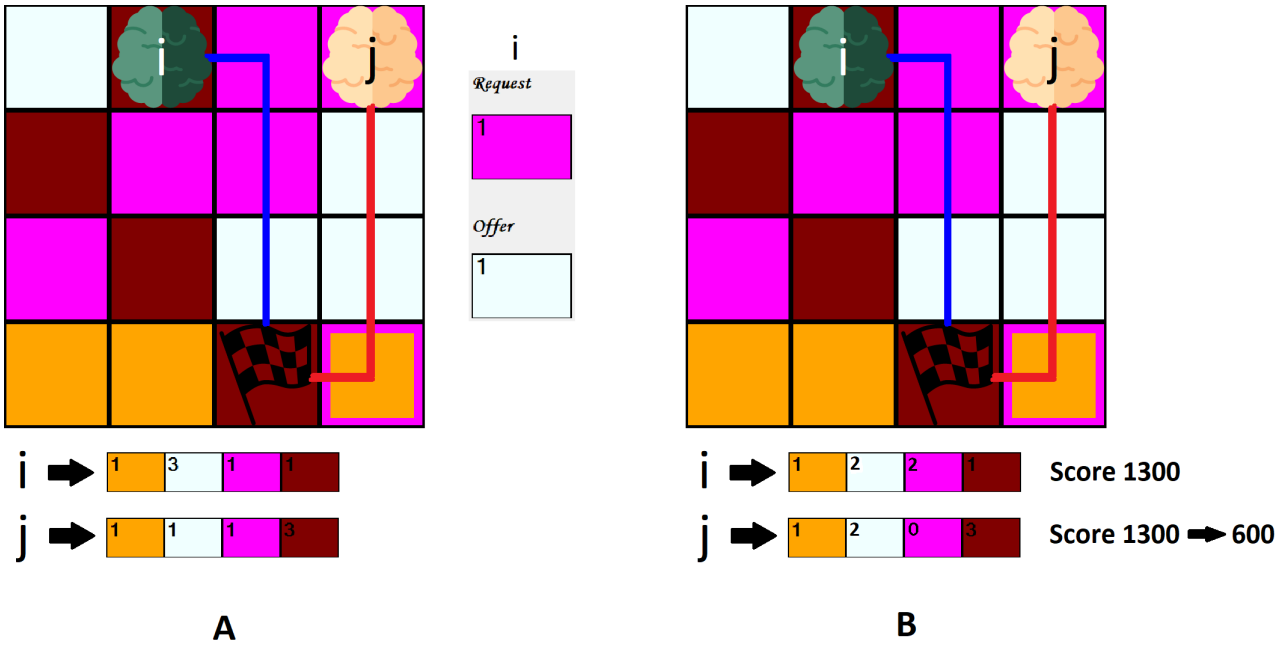


Figure 5.3: A shows the proposal made by agent i, where the requested colour is also the colour the agent concealed. The state in B shows the game after agent j accepted the proposal, receiving a high perceived score and a low actual score after the real colour on the board is revealed.

agent j would also be able to reach the goal, which would make the trade a cooperative one.

The scores are computed after the trade is completed, as shown in Figure 5.3 B. Here the first score shown for agent j is his perceived score, given that the changed colour would have been the actual colour on that tile. The second score indicates the actual score agent j received. Because agent i changed his opponents' beliefs in such way that it only benefits himself, it can be concluded that deception has occurred.

The alternative scenario of the above example is shown in figure 5.4, where agent i did not have a chance to change the tiles on the board. Because now both agents have complete information about the whole game agent i cannot make a deceptive proposal as agent j would not accept it. So the only way to increase his score is by making a cooperative proposal which results in both agents reaching the goal.

Example 2 The following example shows how agent i uses the changed colour to his advantage by making what appears to be a good trade. Figure 5.5 A shows the initial conditions of the game. Concealing the yellow tile next to the goal makes sure that agent j now believes he needs an orange tile instead of a yellow one in order to reach the goal. Agent i does not need a yellow chip in order to reach the goal unlike the previous example, where the agent changed the tile colour because he needed that colour to reach the goal while making sure his opponent could not. Here the agent changes the colour

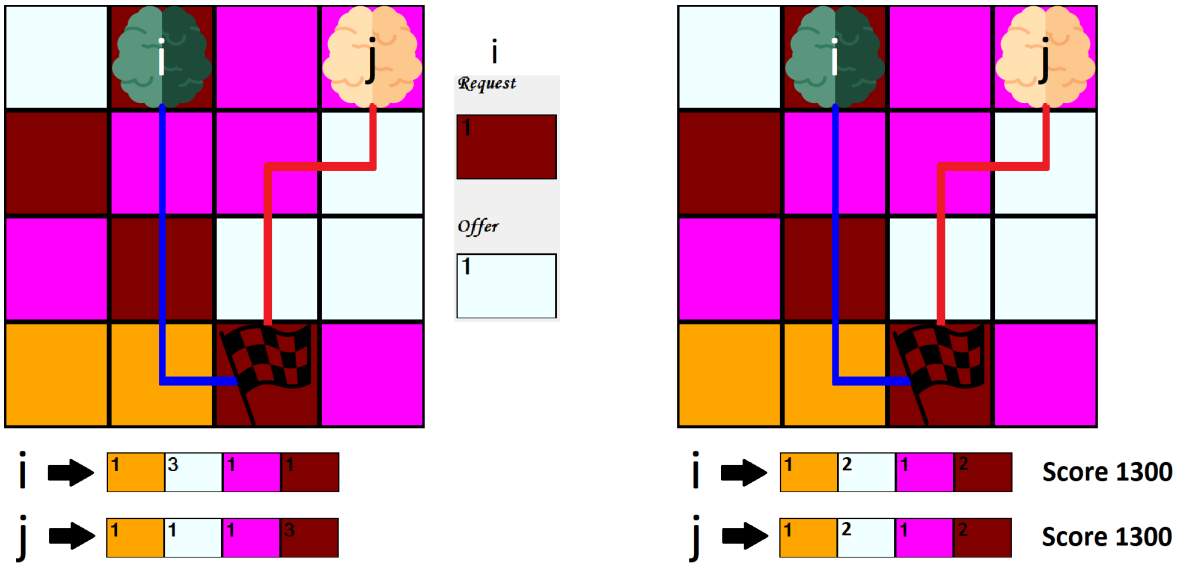


Figure 5.4: Alternative scenario where agent i did not change the tiles on the board. A shows the proposal made by agent i, which is a cooperative move because it enables agent j to reach the goal as well. The state in B shows the game after agent j accepted the proposal and both agents received a high score.

just to make sure that his opponent cannot reach the goal. Because agent i now has to offer an orange chip he is unable to take the most direct path towards the goal so he considers other paths, which can be longer.

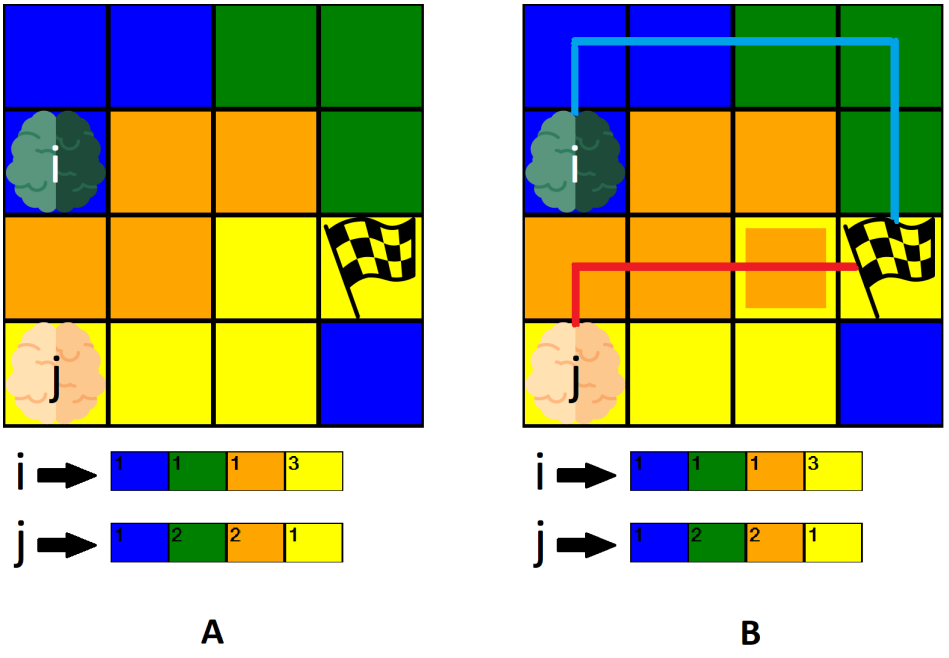


Figure 5.5: Two states of the game where A is the initial board with both players' initial distributions of chips and B is the state of the game after agent i changed the tile. Figure B also shows the paths towards the goal as computed by both agents. Taking a longer path over a shorter one by concealing the appropriate colour is a deceptive move that agent i prefers over the cooperative one

Figure 5.6 shows proposal made by agent i, where he requests three chips in exchange for one orange. Because this offer is not suspicious to agent j, according to the response rules shown in 5.1.4, he will accept it. Because reaching the goal has a bigger priority to agent j than getting a higher score than his opponent he is willing to give more chips in return for less. If he would decline the offer then the score he would receive is significantly less than otherwise. Note that agent i needs to propose such trade that after the actual colours on the board are revealed there would be no new paths possible towards the goal for agent j. Accepting the offer presented in figure 5.6 results in a perceived score of 1200 for agent j and the score of 1400 for agent i. After revealing the actual colours the score of agent j becomes 500 and thus he has been successfully deceived.

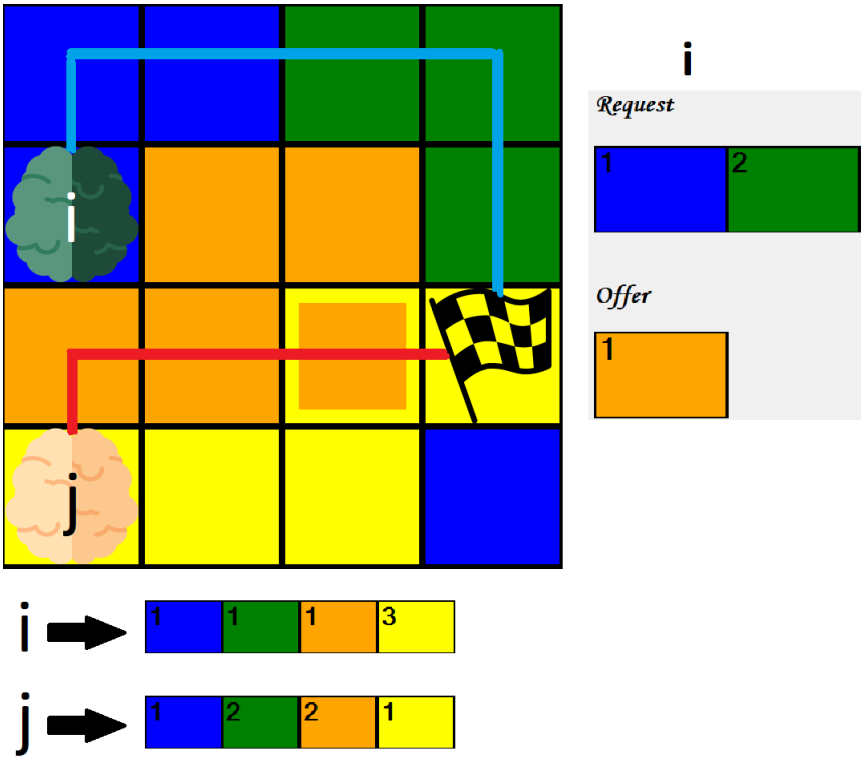


Figure 5.6: Proposal made by agent i after he changed the tile on the board. Here, the agent requests more chips than he offers. Both agents will be able to reach the goal after the proposal is accepted, given the observable board by the agent j

Again, the alternative example where both agents cooperate is shown in figure 5.7. Here agent i makes an almost identical proposal, however instead of offering the orange chip he offers a yellow one. This ensures that agent j is able to reach the goal as well. Note that in this scenario there are multiple proposals possible, one of which is requesting one orange chip and offering two yellow chips. In doing so both agents will be able to reach the goal, however this proposal results in a lower score than the proposal shown above and therefore will not be chosen by agent i.

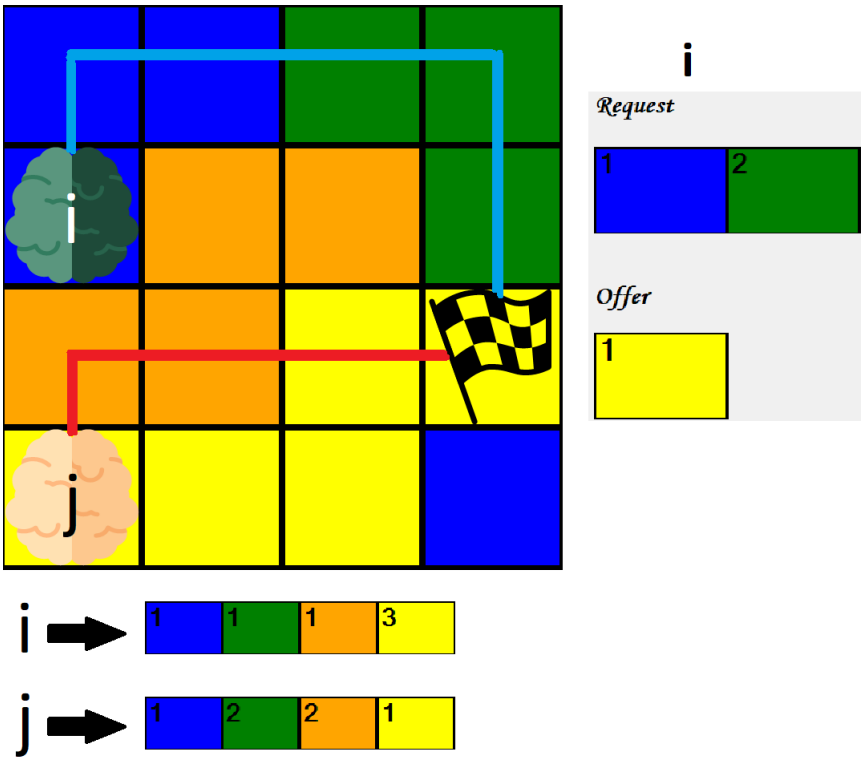


Figure 5.7: Alternative scenario where agent i is not able to change the colours on the board. Here the only way to reach the goal is by cooperation.

5.3 Parameters

This section describes the parameters used for running the simulations. These parameters are ordered according to their implementation functions:

Board

Dimensions of the board are chosen as 4 x 4 where 4 colours are randomly distributed among each tile. Reason for choosing such dimensions and the amount of colours is to reduce the computational resources. The implementation however allows for generalization onto larger boards containing more colours.

Agents

For the purpose of investigating the workings of Theory of Mind agents and interactions between them, both agents use first-order ToM reasoning. In order to find the right offer, a proposer needs to attribute mental content to his opponent which includes his beliefs, goal and intentions. Using this information the proposer can adjust the board in such way that his offer becomes attractive to his opponent. The opponent however needs first-order reasoning too in order to form beliefs about the offer that he has received. If the offer does not coincide with the belief about the environment and proposers' intentions, then the offer is rejected. Reasoning about proposers' intentions implies the need for Theory of Mind beyond the zero-order.

Chip sets

Because the amount of colours in the game is 4, the maximum amount of a certain colour in a chip set becomes 4 as seen in chapter 4.2. The same chip set rules still hold for both agents, where they cannot reach the goal given their initial distributions of chips.

Scoring

Four parameters in the scoring function are defined as follows:

1. *Goal weight*, set to 500 to ensure that both players have incentive to move towards the goal as it increases the score the most.
2. *Distance*, set to -50, the penalty a player receives for each square away from the goal along the Manhattan distance.
3. *Chip weight*, set to 50 to ensure that each player will try to get the largest piece for himself. Otherwise the players would trade all chips that they don't need against just a few that they need.
4. *Base score*, set to 500 to prevent from agents having negative scores.

The above parameters are used to run simulations for two different scenarios, namely one where the agent i is able to change the tiles and another where he cannot change the tiles.

5.4 Simulation Results

For the simulations a 1000 different game boards with agents' chip sets were generated. The agents played on those boards a total of 10 times, making the amount of runs 10000. The decision was made to use this amount of runs in order to minimize the occurrence of coincidental outcomes that might be influential during the analysis. The same amount of runs with same board configurations were performed on a scenario where the proposer could not change the colours on the board, in order to compare the performance of both agents and investigate the significance of having complete information as opposed to incomplete information. Here the proposer is denoted by i and responder by j. For simplicity, the game scenario where the proposer is able to change the tiles is denoted by g_1 and scenario in which both agents have the same amount of information in which proposer could not change the tiles is denoted by g_0. Table 5.1 shows the average scores and standard deviations of all runs for agents playing both scenarios.

	f^i	f^j	σ_i	σ_j
g_0	1274	1265	54.3	47.1
g_1	1299	1036	70.0	307.8

Table 5.1: Average scores f and standard deviations σ agents received when playing a 1000 generated games 10 times for scenarios g_0 and g_1. Both scenarios use the same 1000 boards for consistency.

The results show a slight increase in score for agent i when he can change the tiles on the board. This ability is also associated with a strong decrease in score for agent j. For game scenario g_0, where the agents are cooperating, the values of the scores lie closely together because agent i tries to find proposals that are advantageous for both agents. The reason for a slightly higher average score for agent i is because his ability to make proposals that sometimes result in a higher score even when both agents reach the goal, as seen from the last part of Example 2 in 5.2. A large difference in scores when agents play scenario g_1 can be explained by the difference in information. Because agent i has more information about the board he can use that to his advantage and make a proposal which might seem advantageous for an agent lacking the additional information. The increase in score between g_0 and g_1 is relatively small for agent i because when both agents cooperate they are able to reach the goal and thus receiving a high score. In case of g_1 a deceptive move is not always possible which means that agent i has to make a cooperative move. Board configurations that do allow for deceptive proposals to occur result in agent i reaching the goal, thus receiving equal score if he would cooperate. However, before making a proposal the agent looks for additional routes from his opponent towards the goal. To make sure that his opponent cannot reach the goal in any other way after accepting the proposal, agent i requests additional chips. Upon doing so he needs to fool agent j into thinking that he needs those chips himself in order to reach the goal, which is done by changing more tiles on the board. This leaves agent i with more chips and thus gaining a higher score. The average amount of deceptions for all runs is 0.289 which means that deception occurs on average once in three to four board configurations. Taking the difference in scores between scenarios g_0 and g_1 for agent i and dividing by the amount of deceptions gives an average increase in score per deception of 85.5. Because each chip is set to have a weight of 50 in the scoring function, agent i receives on average 1.7 chips more at each deceptive trade. The average decrease in score per deception for agent j can be calculated in a similar way, by taking the difference in scores for both scenarios and dividing by the amount of deceptions. This gives an average decrease of 762.6 for each successful deception. The reason for such large number is because when a deceptive proposal is accepted the agent j is unable to reach the goal thus receiving a much lower score that he would otherwise. Table 5.1 shows the standard deviations of the scores the agents received during the simulations. For game scenario g_0 the table shows that the scores are tightly spread

around the mean where agent i has a slightly larger spread. The values in this table correlate with the average scores obtained by the agents. The large σ_j value in g_1 is explained by the drastic decrease in score upon a successful deception, which also explains the value for σ_i where agent i receives a slightly higher score than a scenario where no deception has happened.

5.5 Hypotheses Test

The significance of the scores between the proposer and responder is tested by performing t-tests on the obtained data. The subsections below show the hypotheses tests for both environments g_0 and g_1 using the two sample t test. The reason for choosing parametric tests is because the information about the population and the mean of the samples is known, which includes the set of obtained scores of the agents. Additionally the output of the scores follows a distribution, namely a maximum score that can be obtained to a minimum score. Using these assumptions it is expected to obtain more accurate results than using a nonparametric test, which does not assume a certain distribution.

5.5.1 Two sample t-test for g_0

The first significance hypothesis is performed on the scenario g_0 where the proposer is not able to change the tiles initially. The obtained population variances are similar for agents i and j namely $\sigma_i^2 = 2949,65$ and $\sigma_j^2 = 2222,88$, so the t-test performed here will assume *equal variances* using two tailed distribution. As seen from the box plot in Figure 5.8 the data from each sample looks quite symmetric, meaning that a *null hypothesis* H_0 can be stated as follows: $H_0 : \mu_i - \mu_j = 0$. This shows that there is no difference between the scores of agents i and j. For clarification $H_1 : \mu_i - \mu_j \neq 0$ is added which denotes the significant difference between the scores.

The equation below shows the *pooled variance* s^2 obtained from population variances of both agents, with degrees of freedom $df = 1000 - 1$, here denoted as n:

$$s^2 = \frac{(n_i - 1)\sigma_i^2 + (n_j - 1)\sigma_j^2}{(n_i - 1) + (n_j - 1)} = \frac{(1000 - 1) \cdot 2949,65 + (1000 - 1) \cdot 2222,88}{(1000 - 1) + (1000 - 1)} = \quad 2586,29 \tag{5.5}$$

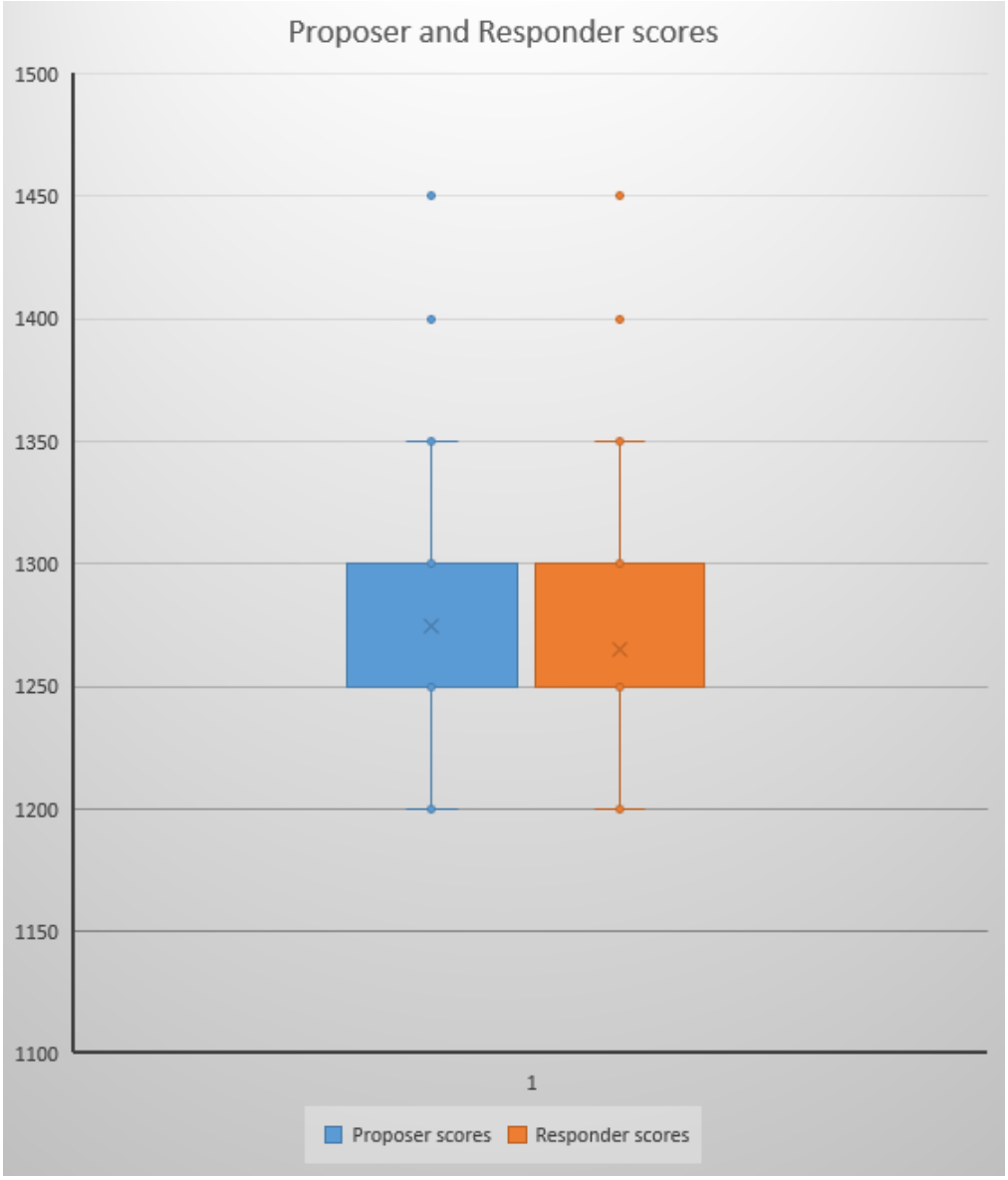


Figure 5.8: Box plot of scores the proposer and responder obtained playing scenario g_0 where no tiles were changed initially

Standard deviation of both sets is obtained from pooled variance as follows: $\sigma = \sqrt{2586,3} = 50,9$. Now the t value is obtained by using the pooled variance, the means and the df:

$$t = \frac{(\bar{x_i} - \bar{x_j}) - (\bar{\mu_i} - \bar{\mu_j})}{s^2\sqrt{\frac{1}{n_i} + \frac{1}{n_j}}} = \frac{(1274,45 - 1264,65) - 0}{2586,29\sqrt{\frac{1}{1000} + \frac{1}{1000}}} = \quad 0,085 \tag{5.6}$$

Taking the alpha level 0,05 (5%) with sample size of 1000 the obtained t-value from the t-value table is 1,962. Using statistical software the p-value can be obtained from the t-value obtained above. The p-value is $0,93 > 0,05$ and t-value $0,085 < 1,962$, so

we accept the null hypothesis H_0 and thus conclude that there is no significant difference between the scores obtained by agents i and j for game scenario g_0.
Note that the above approach considers equal variances. Because the variances of both samples are not quite equal another method can be used, namely the t-test with unequal variances. The equation is shown below, which does not consider pooled variance anymore but only the individual variances:

$$t = \frac{(\bar{x_i} - \bar{x_j}) - (\bar{\mu_i} - \bar{\mu_j})}{\sqrt{\frac{\sigma_i^2}{n_i} + \frac{\sigma_j^2}{n_j}}} = \frac{(1274,45 - 1264,65) - 0}{\sqrt{\frac{2949,70}{1000} + \frac{2222,88}{1000}}} = \quad 4,31 \tag{5.7}$$

In order to obtain the p-value we need to approximate the degrees of freedom using the Welch-Satterthwaite equation[31]:

$$n \approx \frac{\left(\frac{\sigma_i^2}{n_i} + \frac{\sigma_j^2}{n_j}\right)^2}{\frac{\sigma_i^4}{n_i^2\left(n_i - 1\right)} + \frac{\sigma_j^4}{n_j^2\left(n_j - 1\right)}} = \frac{\left(\frac{2949,70}{1000} + \frac{2222,88}{1000}\right)^2}{\frac{54,3^4}{1000^2 \cdot 999} + \frac{47,1^4}{1000^2 \cdot 999}} \approx 1963 \tag{5.8}$$

Now the p-value obtained using the above t-value is $1,71 \cdot 10^{-5} < 0,05$ and the t-value $4,31 > 1,962$ so the null hypothesis H_0 is rejected. Thus according to the unequal variance t-test there is significant difference between the scores obtained by the agents.

5.5.2 Two sample t-test for g_1

The following significance hypothesis is performed on the scenario g_1 where the proposer is able to change the tiles initially. The obtained population variances are dissimilar for agents i and j namely $\sigma_i^2 = 4901,78$ and $\sigma_j^2 = 94744,44$, so the t-test performed here will assume *unequal variances* using two tailed distribution. As seen from the box plot in Figure 5.9 the data from each sample does not look symmetric, so we can assume that the null hypothesis will be rejected. Again H_0 can be stated as follows: $H_0 : \mu_i - \mu_j = 0$. This shows that there is no difference between the scores of agents i and j. Also $H_1 : \mu_i - \mu_j \neq 0$ denotes the significant difference between the scores.

Using the same equation as 5.7 for unequal variances, the following t value is obtained:

$$t = \frac{(\bar{x_i} - \bar{x_j}) - (\bar{\mu_i} - \bar{\mu_j})}{\sqrt{\frac{\sigma_i^2}{n_i} + \frac{\sigma_j^2}{n_j}}} = \frac{(1299,15 - 1044,25) - 0}{\sqrt{\frac{4901,78}{1000} + \frac{94744,44}{1000}}} = \quad 25,54 \tag{5.9}$$

Now approximating the degrees of freedom:

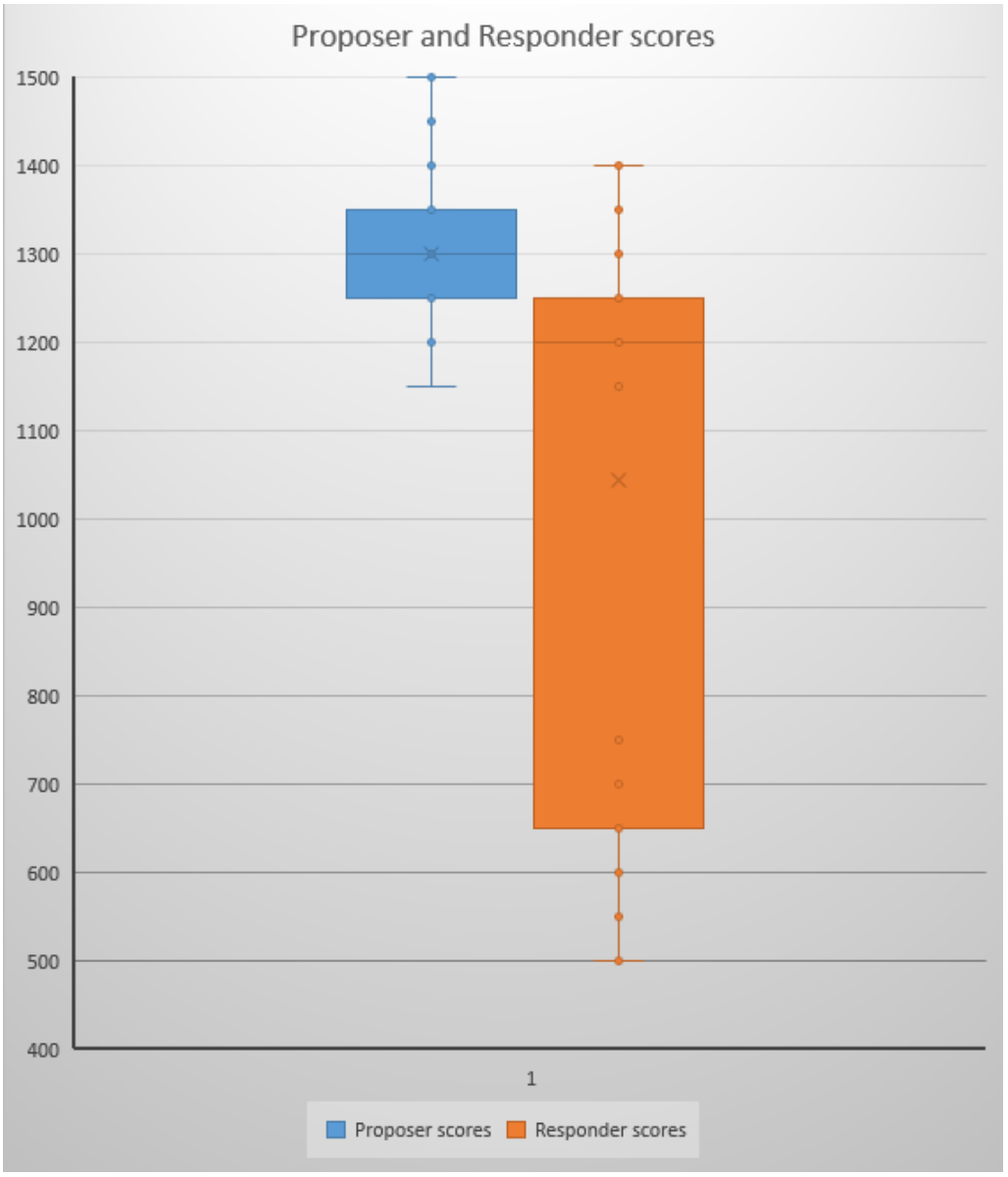


Figure 5.9: Box plot of scores the proposer and responder obtained playing scenario g_1 where tiles were changed initially by the proposing agent i

$$n \approx \frac{\left(\frac{\sigma_i^2}{n_i} + \frac{\sigma_j^2}{n_j}\right)^2}{\frac{\sigma_i^4}{n_i^2(n_i-1)} + \frac{\sigma_j^4}{n_j^2(n_j-1)}} = \frac{(\frac{4901,78}{1000} + \frac{94744,44}{1000})^2}{\frac{70,0^4}{1000^2 \cdot 999} + \frac{307,8^4}{1000^2 \cdot 999}} \approx 1102 \tag{5.10}$$

Because the obtained p-value is $2,17 \cdot 10^{-113} << 0,05$ and $25,54 > 1,962$ we can reject the null hypothesis, as expected. This shows that there is a very significant difference between the scores of agents i and j when playing the scenario g_1.

5.5.3 Significance Test using Pearson Correlation

To show the correlation between the scores obtained by agents a measure called *Pearson Correlation* is used. A positive coefficient would indicate a positive correlation between the agents' scores, meaning that if one score increases the other increases with it as well. From this value it can be shown whether the agents overall behaviour is cooperative or deceptive, where a positive value indicates more cooperation and negative value indicates the agents' tendency for deception. The relevance of correlation decreases the closer the value approaches zero. This coefficient is obtained by taking the covariance of both scores and divide that by the product of the standard deviations. This is given in the equation 5.11.

$$r_{ij} = \frac{\sum f_n^i f_n^j - n\bar{x_i}\bar{x_j}}{(n-1)\sigma_i\sigma_j} = \frac{n\sum f_n^i f_n^j - \sum f_n^i \sum f_n^j}{\sqrt{n\sum (f_n^i)^2 - (\sum f_n^i)^2}\sqrt{n\sum (f_n^j)^2 - (\sum f_n^j)^2}}, \tag{5.11}$$

where $f_n^i = \{f_1^i, f_2^i, ..., f_n^i\}$ are the scores obtained by agent i and similarly f_n^j obtained by agent j. Further n is the amount of observations. The resulting value is always between -1 and 1, where values < 0 indicate a negative linear correlation and > 0 positive linear correlation and zero value indicates no correlation. Both scatter plots representing the correlation of players' scores in environments g_0 and g_1 and the linear trendlines are shown in Figures 5.10 and 5.11 for clarification.

Using Equation 5.11 the value for environment g_0 is obtained: $r_{ij} = 0,65$; and for g_1 results in: $r_{ij} = -0,30$. Positive Pearson correlation coefficient indicates that as one variable increases the other increases as well, whereas a negative value indicates that as one variable increases the other decreases. These results are consistent with the trendlines shown in figures 5.10 and 5.11. Using the Pearson correlation coefficient the t-value can be obtained. The equation that denotes this is shown below:

$$t = \frac{r_{ij}\sqrt{n-2}}{\sqrt{1-r_{ij}^2}} \tag{5.12}$$

Filling in the above equation gives a t-value $27,02$ for g_0 and $-10,10$ for g_1. Taking the alpha level 0,05 (5%) with 1000 degrees of freedom the obtained t-value from the t-value table is 1,962. Because $27,02 > 1,962$ and $0,65 > 0$ we can conclude that the correlation of obtained scores by the agents playing on environments g_0 is significantly positive. For environments g_1 can be shown that $|-10,10| > 1,962$ and $-10,10 < 0$ which indicates that the correlation of obtained scores is significantly negative.

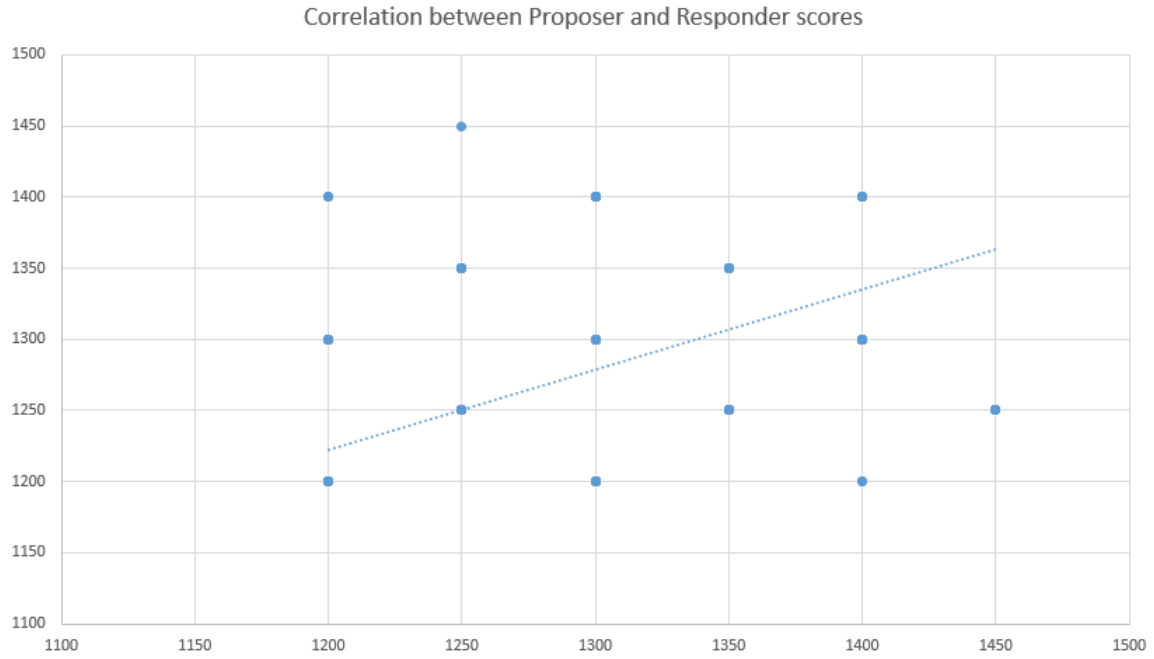


Figure 5.10: Correlation between the scores obtained by agent i and agent j playing on environment g_0

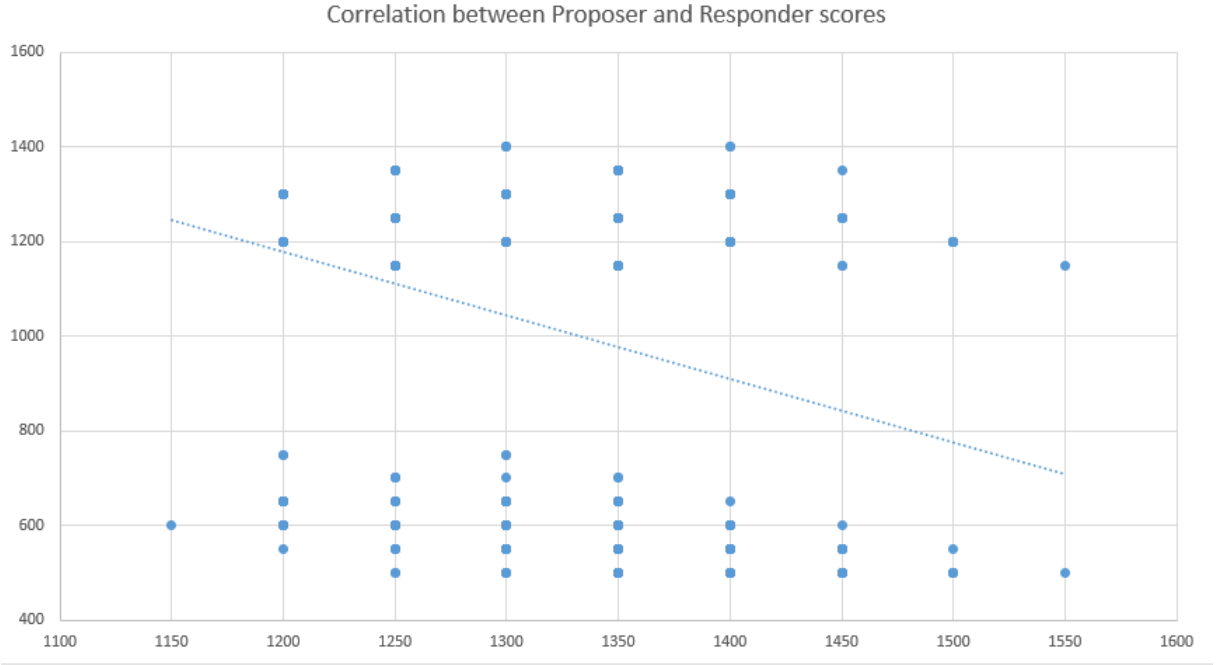


Figure 5.11: Correlation between the scores obtained by agent i and agent j playing on environment g_1

5.6 Analysis of Results

The results and statistical tests above show that the combined scores of both agents are higher when they are playing g_0 games, which can be seen in Figure 5.12. By giving

both agents the same amount of information, the agents cooperate more often, hence the increase in score for agent j playing scenario g_0 as opposed to scenario g_1. However agent i obtains overall higher score through deception than cooperation, supporting the expected Hypothesis 1 H_1. Further, the proposing agent has some advantage over the responder because he is the one that can make a proposal, whereas the responder can only accept a proposal that looks acceptable even if there are better options available.

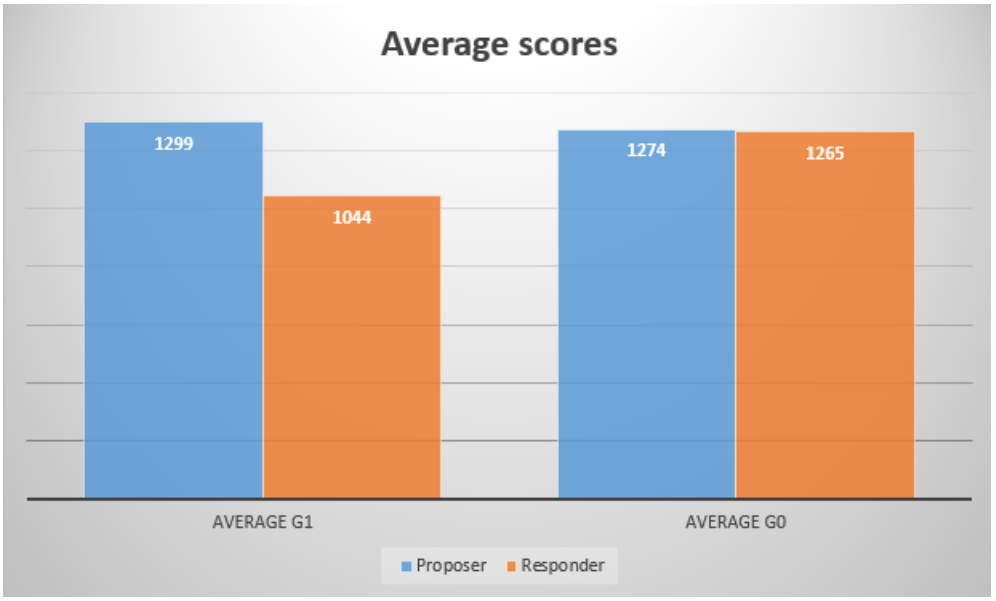


Figure 5.12: Average scores both agents i and j obtained when playing scenarios g_0 and g_1

When the agent i gets an opportunity to hide certain tiles to make a deceptive proposal the standard deviation of agent j becomes very high, as seen from Figure 5.13. This is due to the large difference in scores an agent would receive if he would reach the goal as opposed to if he would not. The standard deviation of the scores of agent i becomes higher as well because in such scenario the agent is able to request more chips from his opponent than needed, sometimes resulting in relatively higher scores. The results shown in the previous section show that an agent having more information has an advantage over the other, supporting the expectation represented by Hypothesis 2 H_2 saying that first-order ToM agents with complete information generally receive a higher score than agents with incomplete information. However in environments where cooperation is expected the advantage an agent has over the other is not as high as one would expect. In this one-shot case the agents have a better chance on reaching the goal and obtain a nearly optimal score by cooperating. While Figure 5.12 shows that agent i obtains a higher average score when playing g_1 games as opposed to g_0, the standard deviation of those scores is higher when that agent is playing g_1 games. This indicates stronger variation in obtained scores, while overall increase in average score is not as high.

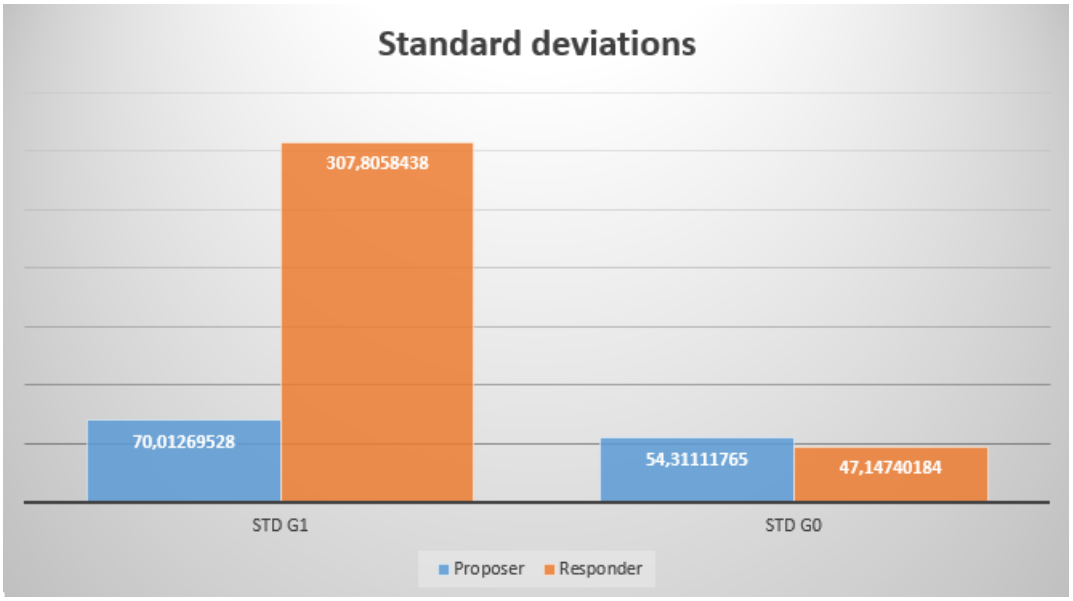


Figure 5.13: Average scores both agents i and j obtained when playing scenarios g_0 and g_1

Using Pearson correlation it is shown that agents playing g_0 scenarios obtain scores that have a positive correlation, which means that as the score of one agent increases the other agents' score increases as well. This indicates cooperative behaviour, which is supported by the agents' average scores for g_0. The correlation coefficient for scores obtained by the agents playing scenarios g_1, however, is negative. This shows the deceptive behaviour of agent i, meaning that when that agents' score increases the other agents' score decreases.

In order to explore deception more in-depth while using different orders of Theory of Mind there needs to be more interaction between the agents during each game. This can be done by making the game a *Repeated Game*, where the agents can interact multiple times with each other during a game or a round. Because interaction should be two-sided the responding agent needs to be able to communicate as well. This can be done by giving the responder the ability to make a counter proposal upon receiving a proposal from the proposer. Because of more frequent interaction both agents can start with the same amount of information, where both of them will have incomplete information. By communicating with each other they would be able to reveal some of the information to each other, either true or false information. This is when cooperation and deception will occur more clearly. By implementing different orders of Theory of Mind a better analysis into deception in environments with incomplete information can be made. This case study is a stepping stone towards the second case study that expands on what is previously stated, which is described in the next chapter.

Chapter 6

Case Study 2: Repeated games

This chapter shows the second approach for investigation of deception in different orders of Theory of Mind agents. The environment used here is configured as a repeated game. In such game scenario the agents interact with each other through continuous alternating offers. More frequent interaction between the agents allows for better understanding of the workings of ToM and how it's used to deceive the opponent. Both agents have incomplete information about the game, which means that none of the agents has an advantage or any prior knowledge over the other agent initially. This incomplete information is in the form of hidden chip sets. Unlike the previous case study, here the agents can assume both roles of being a proposer and a responder during a single game. This allows the agents to get a better understanding of opponents beliefs, desires and intentions. While a proposer always starts the communication at each round, the responder can make a counter proposal upon receiving a proposal. This interaction continues until both agents have reached an agreement and the chips are exchanged or until 20 consecutive offers have occurred without reaching an agreement. The reason for choosing 20 is because the agents rarely come to an agreement after that amount of continuous interactions. Both agents are then advanced one step towards the goal, given that they have the right chips to advance onto the next tile. Then the communication phase starts again, which the initial proposer always starts. Individual scores are computed during each advancement step and the game terminates once either one of the players reaches the goal or once an agent didn't move for 5 consecutive rounds. The next sections describe the model and simulation results for repeated game approach.

6.1 Model

This section shows the model used for this case study, that extends the general model introduced in chapter 4.

6.1.1 Navigation in Repeated Games

Both repeated and one-shot game scenarios implement similar navigation techniques. Because the agents have complete information about each others chip sets in one-shot game, they are able to find a shortest path on the board by combining the given chip sets. The chips that are not in their set can then be requested from the trading partner. In a repeated game the agents initially don't have any information about the other agents' chip set, however this information can be obtained through continuous interaction. Because in repeated games the agents are advanced one step at a time towards the goal, the neighbouring position needs to increase the score of the agent. If it does not, then the agent has no incentive to move. Otherwise, the agent moves onto the tile that will yield him the highest score. In some cases there are movements possible that result in same scores. In such case the agent needs to look further into the movement process by checking the possibly reachable tiles after the tile he is planning on moving onto. The agents are able to look into the movement process as far as their chip set allows. This means that an agent can find a neighbouring tile that will lead him to the goal, given that he has the needed chips in his possession. The position of the tile that results in the highest score will be chosen as follows:

$$Move(pos_i, c_i^t) = \begin{cases} n & if \quad \max\limits_{n \in N} f^i(n, c_i^{t+1}) > f^i(pos_i, c_i^t) \\ pos_i & otherwise, \end{cases} \tag{6.1}$$

where pos_i is the current position of agent i and c_i^{t+1} is his chip set after advancing on the neighbouring tile. If the highest scores are equal then the tile is chosen that has the most neighbouring tiles, ensuring that the agent has the most options later on. This navigation technique is shown in figure 4.1.

6.1.2 Predicting Own Score

Because the agents are moving towards the goal step by step over the period of the game, they need to be able to compute their predicted scores. The agents are able to calculate their position on the board if they were to advance as far as possible towards

the goal given their current chip sets. This enables the agents to see further into the game process. The way this is done is by simulating the next movement processes using the $Move$ function described in the previous subsection, until they cannot advance any further. From this new position n the expected score can be obtained. By using the score function $f^i(s)$ introduced in 4.1 and the Move function, a prediction of agents' i score is calculated as follows:

$$f(c_i^t) = \begin{cases} Move(n, c_i^{t+1}) \rightarrow f(c_i^{t+1}) & if \quad Move(pos_i, c_i^t) \quad \neq \quad pos_i \\ f^i(pos_i, c_i^t) & otherwise \end{cases} \tag{6.2}$$

Here, $Move(n, c_i^{t+1}) \rightarrow f(c_i^{t+1})$ denotes the one step the agent makes towards the goal followed by the above score function. The current position pos_i then becomes the next position n. This recursive behaviour ensures that the agent tries to see how far he can advance towards the goal before calculating his score. After the negotiation process the agents adjust their expected scores based on their new distribution of chips.

6.1.3 Agents

This game scenario considers the agents to have the same amount of information initially. The difference in information later on during a game depends on the agents' ability to observe and reason about the game state. Initially, both agents are set to have incomplete information about each others preferences. This is done by obscuring their chip sets from each other. By reasoning about the possible chip sets the opponent may have, an agent is able to make an offer that might be deceptive, cooperative or just advantageous for himself. Forming beliefs about possible chips that the opponent may have is achieved by interacting with each other through continuous alternate offers. Unlike one-shot scenario, in a repeated game both agents can make offers and respond to offers. This enables the agents to communicate their intentions through actions to each other continuously, giving the responding agent j a better chance to communicate with the proposer i, unlike the one-shot scenario. Whether it results in a cooperative action or a deceptive can be seen in the changes that occur in *zero-order beliefs* and *expected values*, more on that in sections 6.1.4 and 6.1.6.
Each trade can only include one chip that is being requested and one chip that is being offered. The responding agent can either accept, decline or make a counter offer. In case of the latter option, the roles of the agents are reversed. Because the agents have no information about each others chip sets initially, the agent chooses a trade that will yield him the highest score, not considering whether the trade will be accepted by the

other agent. While the game progresses, the agents get a better understanding of each others chip sets and the proposals to do a trade become more constrained.

6.1.4 Zero-Order Beliefs

Initially, the zero-order ToM (ToM_0) agent needs to form beliefs about whether the opponent will accept a certain proposal. A way of achieving this is to make a list of possibilities about the likelihood of the opponent accepting a certain offer for each chip set that could possibly be owned by him $b^{(0)} : \mathbb{C} \rightarrow [0, 1]$, as presented by De Weerd et al.[26] as zero-order beliefs. Building upon this idea a belief matrix is made that can be represented for agent i as $b_i^{(0)mn} : \mathbb{C} \rightarrow [0, 1]$. Here m represents a requested chip and n one offered chip $\{m, n\} \in C$, where C is the set of all colours in the game, for each possible chip set. In this matrix each column shows the requested chip and each row shows the offered chip. Because opponents chip set is unknown during the first steps, the agents assign equal probabilities to each proposal. These probabilities depend on the opponent having the requested chip in his chip set. This probability can be computed with $1 - \frac{1}{(w \cdot h)/c+1}$, where the denominator is the maximum amount of each colour in a chip set as shown in 4.2. Given a 4×4 board of 4 colours, the probability of the opponent having a certain chip in his chip set becomes: $1 - \frac{1}{5} = 0.8$. An initial zero-order belief matrix is shown in Table 6.1, where the letters R, G, B and Y indicate the colours red, gree, blue and yellow that make up the game. The updated belief matrix based on the observations of an agent and his beliefs about which chips the opponent has in his possession is shown in Table 6.2.

	R	**G**	**B**	**Y**
R	0.00	0.80	0.80	0.80
G	0.80	0.00	0.80	0.80
B	0.80	0.80	0.00	0.80
Y	0.80	0.80	0.80	0.00

Table 6.1: Zero-order belief matrix of likelihoods that a certain proposal will be accepted, as initially formed by the agent. The rows indicate a colour that the agent requests and columns a colour that the agent is willing to offer. Because the agent has no information about opponents' chip set, he makes assumptions about the likelihood of the opponent having certain chips.

Because zero-order beliefs do not depend on the mental content of others, an agent using only these beliefs will not consider the beliefs and intents of his opponent and will only try to increase his own score. Values in the belief matrix are updated during interaction with other agent and the observation of the game state. If the agent believes

that a certain chip is not present in opponents chip set then the zero-order beliefs are updated accordingly. This means that the agent will eventually learn that if a certain chip is not present in opponents set, the opponent will not accept a proposal that requests that chip.

6.1.5 Beliefs about Opponents Chips

During each interaction the agents update their beliefs about the chips that are owned by the other player, from now on named as *chip set beliefs* for simplicity of naming. During the start of the game at communication round $t = 0$ the agent i enumerates all possible chip sets, here denoted as $\mathbb{C}_i^t = \mathbb{C}$. Each agent considers that every possible chip set might be opponents chip set where each chip set initially has a likelihood $l_i = \frac{1}{|\mathbb{C}_i^t|}$ of being the right one, where $|\mathbb{C}_i^t|$ denotes the length of the set. The maximum amount of each colour in a given chip set is $\frac{(width \cdot height)}{|C|}$. The amount of chip sets at round $t = 0$ that result after enumeration becomes:

$$|\mathbb{C}_i^t| = \left(\frac{(w \cdot h)}{|C|} + 1\right)^{\frac{w \cdot h}{|C|}}, \tag{6.3}$$

where $|C|$ is the amount of colours on the board, w and h are width and height dimensions of the board respectively. The reason for adding one to the base of the exponent is because a given chip set can contain zero values for each colour. As an example, given a game scenario where the board is 4×4 consisting of 4 colours, the amount of possible chip sets that a player can have is $5^4 = 625$. Based on these chip set beliefs, an agent can make assumptions about the possibility of his offer being accepted. Subsequently the agent can also make approximations of his score given a requested or accepted proposal.
Agents update their chip set beliefs regularly throughout the game. These update steps are given in 6.1.8. The chip set beliefs influence the zero-order belief matrix directly. Consider a case where two agents i and j play, where agent i has a zero-order belief matrix as shown in Table 6.1. After an amount of interactions agent i believes that agent j does not have a green chip in his possession, based on his chip set beliefs. He then proceeds to update his zero-order belief matrix to include this new belief. The new belief matrix is shown in Table 6.2.

6.1.6 Expected Values

In order to find proposals that will yield a higher score the agents compute expected values for each colour combination of chips possible. Again, like in a zero-order belief

	R	**G**	**B**	**Y**
R	0.00	0.80	0.80	0.80
G	0.00	0.00	0.00	0.00
B	0.80	0.80	0.00	0.80
Y	0.80	0.80	0.80	0.00

Table 6.2: Zero-order belief matrix after some interaction between the agents has occurred. The agent that builds up this matrix believes that his opponent does not have any green chips, resulting in each proposal that requests a green chip from the opponent having a likelihood of 0.00 of being accepted

matrix, this results in a matrix of all possible trades of chips. The expected values in the matrix correspond to the score the agent would receive after his proposal is accepted by the opponent and the agent is advanced as far as possible towards the goal. Computing this value requires the probability of opponent accepting the proposal, which can be found at the same position in zero-order belief matrix. Agent i computes the expected value $EV_i^{(0)mn}$ at communication round t as follows:

$$EV_i^{(0)mn}(c_i^t, b_i^{(0)mn}) = b_i^{(0)mn} \cdot f(c_i^{t+1}) + (1 - b_i^{(0)mn}) \cdot f(c_i^t), \tag{6.4}$$

where $b_i^{(0)mn}$ is the zero-order belief likelihood corresponding to the proposed chips in the expected value matrix and $f(c_i^{t+1})$ denotes the predicted score agent i would receive if his trading partner would accept the proposal.

6.1.7 Interaction Between Agents

During the game both agents can assume the role of a proposer and responder, however the one that is assigned the initial role of a proposer always starts after the players are advanced one step towards the goal. After the initial proposer made an offer, the responder has an opportunity to make a counter offer instead of accepting or declining it. If that is the case then the roles of the agents are reversed for the duration of that interaction. The above proposal and response functions are explained more in-depth below.

Making a proposal When making an offer, the proposing agent i selects the highest expected value in his expected value matrix. The row and column in the expected value matrix correspond to the colours of requesting and offering chips that are chosen as a

proposal. If the expected value matrix contains multiple highest expected values then the agent selects one at random and makes the corresponding proposal p_i^{best}:

$$Propose_i(p_i^{best}) \xleftarrow{R} \underset{\{m,n\} \in C}{argmax} \quad EV_i^{(0)mn}(c_i^t, b_i^{(0)mn}) \tag{6.5}$$

An example expected value matrix is shown in Table 6.3. Here the agent looks for the highest value, which is 800. This value corresponds to the row representing a colour that the agent should request and a column representing a colour that the agent should offer, therefore the agent will make a proposal requesting the colour blue and offering the colour yellow.

	R	**G**	**B**	**Y**
R	450.0	600.0	600.0	620.0
G	480.0	450.0	520.0	450.0
B	450.0	720.0	450.0	800.0
Y	450.0	520.0	450.0	450.0

Table 6.3: Expected value matrix as computed by an agent. Here the rows represent the colour that the agent requests and columns represent the offered colours. The values are the scores as predicted by the agent. The agent will choose the highest value in the matrix when making a proposal or responding to a proposal.

Responding to a proposal Upon receiving a proposal p_i, the responding agent j determines whether his maximum expected value is higher than the score he would receive if he were to accept the offer. The score $f(c_j^{t+1})$ that is computed by agent j if he were to accept the given proposal p_i^t is denoted here as $p_i^t \rightarrow f(c_j^{t+1})$. If that score is higher than his current score and at least as high as the best expected value, then the offer is accepted. If highest expected value is larger than the score agent j would receive, he can make a counter offer. That means the agent j knows an offer that would benefit him more than the current offer proposed by i. This response function is done in a similar way as presented in the model of De Weerd et al.[26], however the agents here consider every chip set likelihood in their chip set beliefs before making the following response:

$$Response_j(p_i^t \rightarrow f(c_j^{t+1})) = \begin{cases} Propose_j(p_j^{best}) & if \quad EV_j^{(0)best}(c_j^t, b_j^{(0)*}) > f(c_j^t) \quad and \\ & \quad EV_j^{(0)best}(c_j^t, b_j^{(0)*}) > p_i^t \rightarrow f(c_j^{t+1}) \\ accept & if \quad p_i^t \rightarrow f(c_j^{t+1}) > f(c_j^t) \quad and \\ & \quad p_i^t \rightarrow f(c_j^{t+1}) \geq EV_j^{(0)best}(c_j^t, b_j^{(0)*}) \\ decline & otherwise \end{cases} \tag{6.6}$$

In the equation above the zero-order belief $b_j^{(0)*}$ corresponds to the same position in the expected value matrix EV_j^{best}. The agent makes a counter proposal p_j^{best} if he considers that it will result in the highest score among his current score $f(c_j^t)$ and the score he would obtain if he were to accept the offer $p_i^t \rightarrow f(c_j^{t+1})$.

6.1.8 Updating Chip Set Beliefs

The chip set beliefs are updated several times at round t according to the following rules:

- **After received a proposal**, the other players' offered colour should occur at least once in his chip set.
- **After received a proposal**, the requested colour by the other player cannot have an upper bound value in his chip set.
- **After a trade has been accepted**, the colour that is given to the other player needs to be present in his chip set at least once.
- **After a trade has been accepted**, the colour that the other player gave cannot have an upper bound value in his chip set.
- **After both players advanced towards the goal**, the other player cannot have an upper bound value of the colour where he is moved to.
- **After both players advanced towards the goal**, if the other player did not move then the neighbouring colour does not occur in his chip set.

The chip set beliefs that do not meet the above requirements have a zero likelihood of being true at round t.

6.1.9 Updating Beliefs

Agents update their beliefs depending on the actions of the opponent and their role in the negotiation process. The update steps are explained below for each role.

Responder After receiving a proposal, the responder makes the following updates:

- *Update chip set beliefs:* The beliefs of an occurrence of a certain colour in opponents chip set are updated accoording to the rules described in 6.1.5.
- *Update all zero-order beliefs:* Because the chip set beliefs are updated, the beliefs of a proposal being accepted by the opponent need to be updated as well. Using this update the agent gains a better understanding of which offers are less likely to be accepted, because the opponent lacks the required chips.
- *Update one zero-order proposal belief:* The belief likelihood of opponent accepting exactly the same proposal that is made by him need to be lowered. This is intuitive because if the opponent assigns a high value to the chip he requests, it is unlikely he will accept a trade that requests this chip from him. Similarly, if the opponent offers a certain chip, that means that he most likely does not need it.
- *Update Expected Values:* Updating the expected values is necessary in order to make a response based on the beliefs that have been updated previously. This enables the agent to make a prediction of his score early on in the negotiation.

Proposer Depending on the response the proposer received from his trading partner, he updates his beliefs. After receiving an *accept* response from his trading partner, the proposer makes the following updates:

- *Update chip set beliefs:* The beliefs of an occurrence of a certain colour in opponents chip set are updated according to the rules described in 6.1.5.
- *Update all proposal beliefs:* Same as for the Responder because the chip set beliefs are updated, the beliefs of a proposal being accepted need to be updated as well.
- *Update one proposal belief:* Lower the belief likelihood of opponent accepting a proposal that offers the chip he just gave and requests the chip he just received.

If the offer is declined then the proposer cannot make assumptions about opponents chip set and only one update will take place:

- *Update one proposal belief:* Lower the belief likelihood of opponent accepting exactly the same proposal in the future.

If the Responder can't make a counter offer then both agents are advanced one step towards the goal. In case of such event, both agents update their chip set beliefs accordingly:

- *Agent i update chip set beliefs:* If agent j didn't advance towards the goal, then it means that the colours of his neighbouring squares that increase his score are not present in his chip set.
- *Agent i update chip set beliefs:* If agent j did advance towards the goal, then it means that he cannot have a maximum amount of the colour he advanced on in his chip set.

The above update steps also hold for agent j.

6.1.10 First-Order Theory of Mind Agent

The previous subsections describe the behaviour of Zero-Order ToM agents. First-Order ToM (ToM_1) agents consider the opponent to have beliefs and intents as well. So in addition to his zero-order proposal beliefs $b^{(0)}$ he also has a set of first-order beliefs $b^{(1)}$. These beliefs are initially formed for each possible opponents chip set, resulting in a list of matrices as shown in Figure 6.1. This enables a ToM_1 agent to see the game from the perspective of his opponent to a certain extent. By going through the list of belief matrices the agent can reason about the likelihood the opponent thinks that ToM_1 agent will accept a certain offer.

The amount of belief matrices in such list decreases with the chip set beliefs while the agent gets a better understanding about opponents chip set. So the agent i considers the possibility that the opponent j believes that the probability of him accepting the proposal $p_j^t = mn$ is $b_i^{(1)mn}$. Using his first-order beliefs a ToM_1 agent can estimate the expected values in order to predict his opponents behaviour. This is done by reasoning about whether the opponent will accept a certain proposal. Using the ToM_0 response function, ToM_1 agent can find out if the opponent will accept the offer or make a counter offer. If he would make a counter offer then the agent compares whether the outcome of accepting such an offer will increase his own score. A more in-depth explanation on first order expected values is given in section 6.1.10.1 below. While still considering his expected values which are obtained using zero-order proposal beliefs the agent makes a

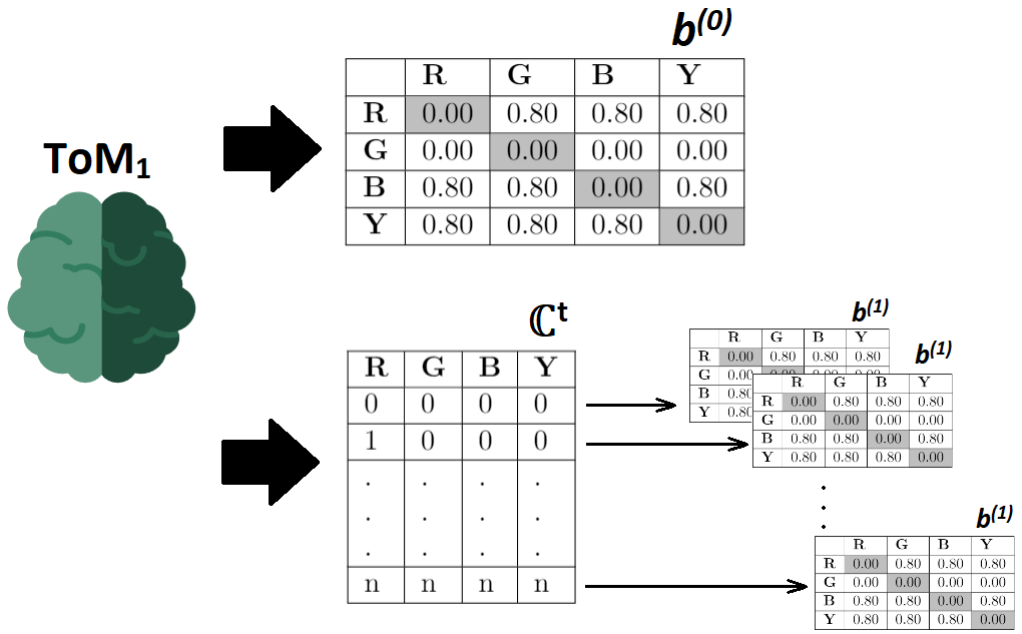


Figure 6.1: First-order *ToM* agent forms besides his zero-order beliefs also a set of first-order beliefs based on his belief about opponents chip set

consideration between those and the new expected values, depending on his *confidence* in the model of his opponent[26]. The next sections will expand on the above process.

6.1.10.1 First-Order Expected Values

Using his first-order beliefs $b^{(1)}$ a ToM_1 agent is able to make predictions of his opponents' behaviour by using his response function. This is done by first calculating the expected values as computed by the opponent $EV_{j'}^{(0)}$. The agent does so by assuming that each chip set $c_{j'} \in \mathbb{C}_i^t$ in his chip set beliefs could be his opponents', each with a certain likelihood that is being continuously updated based on his observations of the game and interactions with the opponent. The opponent as modelled by agent i is here denoted as j'. The agent then computes the expected value for an opponent given that a chip set $c_{j'} \in \mathbb{C}_i^t$. This expected value is here denoted as $EV_{i*}^{(1)mn}$. This expected value depends strongly on the response that the modelled agent j' makes which is based on his chip set $c_{j'}$, expected value $EV_{j'}^{(0)}$ and the proposal p_i^t. For each possible proposal the ToM_1 agent i computes the expected value as follows:

$$EV_{i*}^{(1)mn}(p_i^t, EV_{j'}^{(0)}, c_{j'}) = \begin{cases} f(c_i^{t+1}) & if \quad Response_{j'}(p_i^t \to f(c_{j'}^{t+1})) = accept \\ f(c_i^t) & if \quad Response_{j'}(p_i^t \to f(c_{j'}^{t+1})) = decline \\ \max\{p_{j'}^{t+1} \to f(c_i^{t+1}), f(c_i^t)\} & otherwise, \end{cases} \tag{6.7}$$

where $p_i^t \rightarrow f(c_{j'}^{t+1})$ denotes the score agent j' would receive given the proposal p_i^t and $p_{j'}^{t+1} \rightarrow f(c_i^{t+1})$ is the score agent i would receive if he would accept the counter proposal $p_{j'}^{t+1}$ made by agent j'. Note the max function in the above equation, which allows the agent i to compare the score he would receive with his current score. The above equation shows that a ToM_1 agent is able to make predictions about the future behaviour of his trading partner by using the response function of the modelled agent j'. Applying the above function to each chip set in the agents' chip set beliefs $\mathbb{C}_i^t$ results in a list of expected value matrices that are based on each separate possible chip set. Taking the average of those matrices gives an estimated expected value matrix $EV_{i**}^{(1)}$ given the agents chip set beliefs.
Although using the above obtained expected value implies first-order theory of mind reasoning, the ToM_1 agent considers the possibility that his first-order beliefs $b^{(1)}$ may be incorrect or not accurate enough to predict opponents behaviour. If this is the case then the agent can use his zero-order beliefs instead. This is done by defining a *confidence variable* $v^1 \in [0, 1]$ that shows the confidence the agent has in his first-order theory of mind predictions of scores. The agent computes weighted expected values based on his confidence between his zero-order expected values and the estimated expected values given his chip set beliefs as follows:

$$EV_i^{(1)mn}(b_i^{(0)mn}, b_i^{(1)mn}, \mathbb{C}_i^t, v^1) = v^1 \cdot EV_{i**}^{(1)mn} + (1 - v^1) \cdot EV_i^{(0)mn} \tag{6.8}$$

Because the agents update their beliefs continuously throughout the game, for example upon receiving a proposal, a ToM_1 agent takes such actions into account. Before making a proposal the agent first simulates the updates his trading partner would do if he would receive that proposal. This allows the agent to reason about what the other agent would do when receiving different proposals. The next subsection describes additional updates that are made by a ToM_1 agent.

6.1.10.2 Updating Beliefs

A ToM_1 agent makes the same updates as a ToM_0 agent would, however such agent is also capable in updating his first-order beliefs. These beliefs are updated at the same time when the opponents belief updates occur.
When updating his chip set beliefs, the ToM_1 agent first uses his zero-order update method in order to decrease the amount of chip sets that could possibly be opponents'. Subsequently the agent then computes the expected value the opponent would get given a possible chip set. If the best expected value, given this chip set, is higher than proposers expected value then the likelihood of this chip set being the right one is lower. If there

is a chip set that yields the opponent the highest expected value given his proposal, then the likelihood of it being the right one is higher. Because there exist multiple chip sets that yield the highest expected value for a certain proposal, multiple continuous interactions are needed in order to narrow down the search space.

6.1.10.3 Updating Confidence

The confidence variable v^k is updated after chip set beliefs are updated, upon receiving a proposal. The agent looks for each chip set that he assigns a belief likelihood higher than 0, whether the proposal yields the highest expected value. Possible opponents' chip sets that will yield the highest expected value given that the proposal is accepted increase the confidence in the model. This confidence increases with the accuracy of such possible chip sets. Some possible chip sets yield a lower expected value than a best expected value, given some proposal. Such chip sets do not increase the confidence as much, however still by some degree. This confidence variable allows an agent to get an idea about how well the order of Theory of Mind he is reasoning at fits the opponents' behaviour. The rate at which the agents update their confidence is also influenced by a learning speed parameter $\lambda \in \{0, .., 1\}$. This constant parameter regulates how fast the agent adapts to the opponents behaviour. The value of learning speed is left at $0,5$ throughout the research. The model for confidence is given by 6.9, where j' is one of the possible models for agent j believed by agent i and $l_{j'}$ is the belief likelihood of a certain chip set $c_{j'}$ being the right one.

$$v^k = (1-\lambda) \cdot v^k + \lambda \cdot \sum_{c_{j'} \in \mathbb{C}_i^t} l_{j'} \cdot \frac{EV_{j'}^{(k)mn}(c_{j'}, b_{j'}^{(k)mn})}{\max EV_{j'}^{(k)}(c_{j'}, b_{j'}^{(k)})} \tag{6.9}$$

Here mn indicate the proposal made by agent j and k represents the order of Theory of Mind the agent updating his confidence in.

6.1.11 Measuring the Occurrence of Deception

Because this case study depends on the emergence of deceptive behaviour in agents through continuous play, a measure that captures such behaviour needs to be defined. Here change in a players' belief can occur in two different forms, both of which influence his behaviour. However in order for successful deception to occur, both changes in an agents' beliefs need to occur where one follows from the other. The first measure relies on the changes occurring in an agents' chip set beliefs $\mathbb{C}^t$. Such agent would be considered as *deceived* if he fails to approximate his opponents' chip set leading to a lower expected

value. This could happen through different combinations of proposals that will make the agent believe that a certain chip could not possibly be contained in his opponents' chip set. By looking at the list representing the chip set beliefs of an agent prior to communication step and after, a comparison of both lists can be made while searching whether the actual opponents' chip set is occurring within those lists. If the chip set is not inside the list then the beliefs of that agent are successfully changed.
The second measure looks for changes occurring in agents' expected value matrices. The best expected values of both agents are determined prior to the communication step and are compared to the values after the communication. If an agent would attain a higher expected value after the communication step than before, while his opponent has a lower value than before because of the change that has occurred in his chip set beliefs, then it can be considered that some deceptive behaviour has occurred. Similarly because of that change in his chip set beliefs the agent will choose a different action from his expected value matrix than he otherwise would, resulting in a successful deception. A diagram representing the above described process is shown in Figure 6.2 for clarification. Here agent i deceives agent j by changing that agents' beliefs about his chip set after the communication step. Because of the changed chip set beliefs agent j now computes his best expected value, which is lower than before.

Combining both measures as explained above makes sure that the changes that occur in an agents' expected value matrix did not happen because of incorrect information that is being corrected. Note that an agent does not necessarily have to win the game in order for deception to occur, or an opponent that is deceived does not necessarily lose the game as some incorrect beliefs only force an agent to look for alternative moves.
Cooperative moves are being measured using the same idea as the second measure described above. The difference with cooperative moves is the best expected values of both agents need to be either higher or lower than before the communication step. Once both values increase or decrease at the same time it can be considered a cooperative move, because the agents are agreeing on some middle-ground and choosing for lower scores or both agents are working together to obtain the highest values possible.

6.2 Examples

The following paragraphs show two examples of repeated games played by a proposing agent i and responding agent j. First example shows how both agents are using zero-order ToM in order to make and respond to proposals, and try to reach the goal. Second example shows a first-order ToM agent i playing against a first-order ToM agent j. In both cases either cooperation or deception is expected, however which one cannot

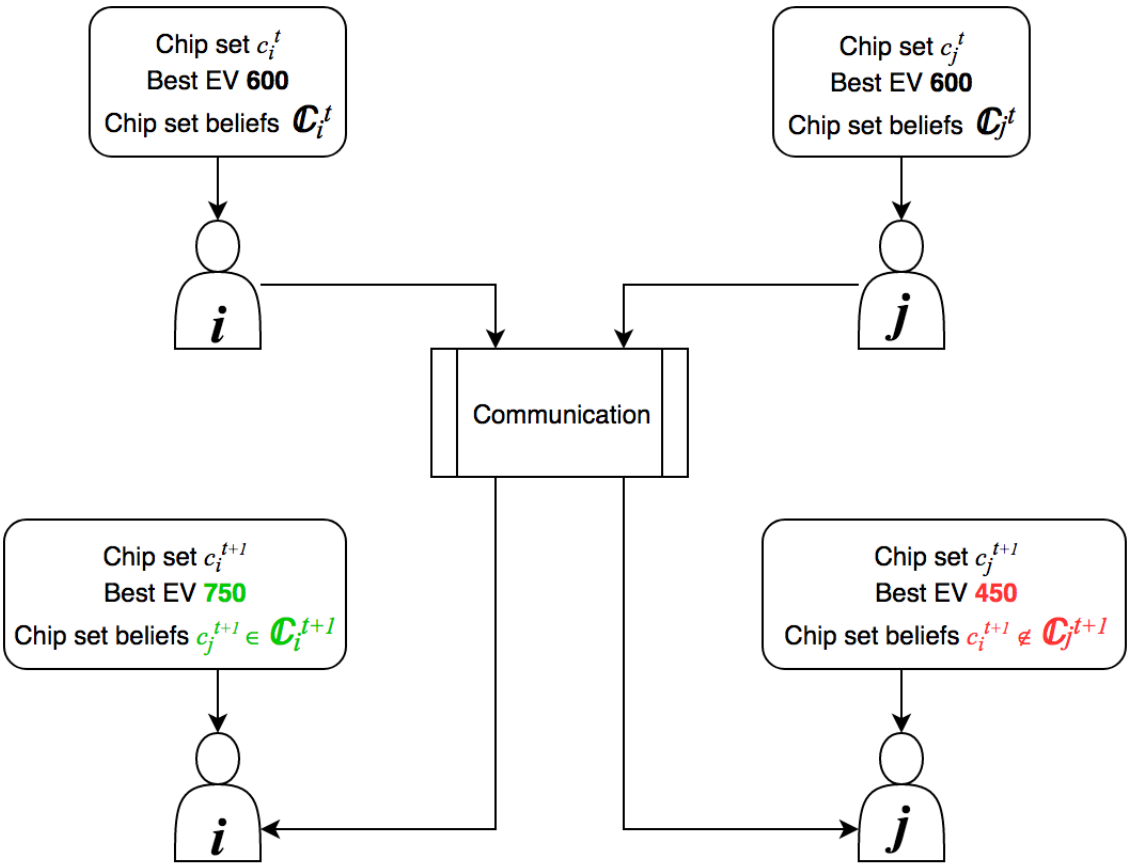
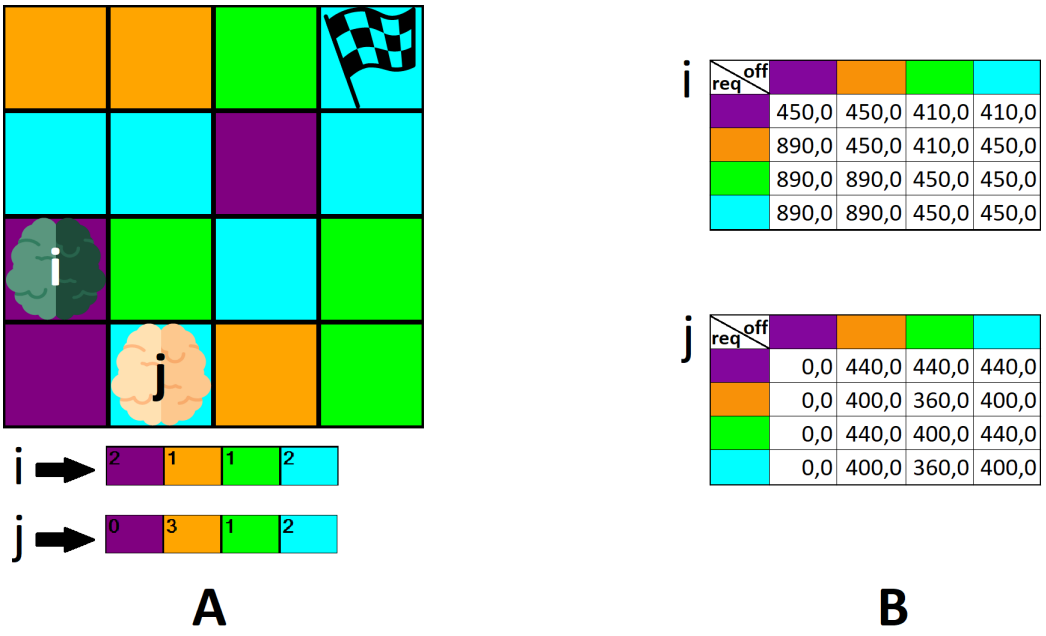


Figure 6.2: Diagram showing the beliefs of the agents before and after communication. Here agent i successfully deceives agent j

be predicted in advance. This is because it depends on multiple factors such as the distribution of colours on the board, distribution of chips both players own and the proposals and responses both agents make. Both scenarios are configured in such way that the players cannot reach the goal without trading, and there are enough chips to make sure both agents reach the goal simultaneously if they are willing to cooperate.

Example 1 Consider the board layout shown in Figure 6.3 A, where agent i is the proposer and agent j is the responder. Both agents cannot view each others' chip sets and therefore assume initially that every possible distribution of chips $\mathbb{C}$ has a chance of being opponents' set. Using this assumption both agents are able to construct their zero-order belief matrix as shown in 6.1.4. Initially, this matrix contains only values of 0,8 as explained in that section. Using the obtained values representing the likelihoods that certain proposals will be accepted the agents compute their expected values. Those values represent the score as predicted by the agents, in the form of a matrix. The method the agents use to compute expected values is shown in 6.1.6. The resulting expected value matrix for both agents is shown in Figure 6.3 B. The matrix created by



i

req \ off				
	450,0	450,0	410,0	410,0
	890,0	450,0	410,0	450,0
	890,0	890,0	450,0	450,0
	890,0	890,0	450,0	450,0

j

req \ off				
	0,0	440,0	440,0	440,0
	0,0	400,0	360,0	400,0
	0,0	440,0	400,0	440,0
	0,0	400,0	360,0	400,0

Figure 6.3: Figure A shows the initial board layout with both players' initial chip sets. Figure B shows both expected value matrices that agents i and j computed. Here the rows represent the colours the agent requests and columns represent the colours the agent offers.

agent i shows that the agent predicts a score of 890 if he would request any colour while offering purple or orange. By looking at figure A it can be seen that agent i is able to reach the goal if he would receive one chip of a colour other than purple. The reason for the zero values along the purple column in the matrix created by j is because the agent cannot offer any purple chips as they are not in his chip set.

Agent i then randomly selects the best expected value from his matrix and makes a matching proposal as shown in Figure 6.4 A. In his proposal agent i requests an orange and offers a purple chip, which has the highest expected value in the matrix of agent j. This can be seen by reversing the proposal, resulting in a request for a purple chip and offer of an orange chip by agent j. The matching value in his matrix is 440, which is the highest in his matrix, and thus agent j accepts the offer. This can be seen in 6.4 B, where the agents completed the exchange and are advanced one step towards the goal along their respective shortest paths and obtaining a score of 300. Before and after the exchange both agents update their zero-order beliefs and their expected values. The changed matrices after the agents are advanced are shown in Figure 6.5. Here agent i is sure that he will reach the goal if he does not give away his green and cyan chip, whereas the agent j does not believe he will reach the goal if he is going to trade any chips but orange. By looking at the Figure 6.4 B it can be confirmed that agent j needs all but orange chips to reach the goal. Because agent i has offered a purple chip during the first communication round, agent j now believes that purple chips are less important

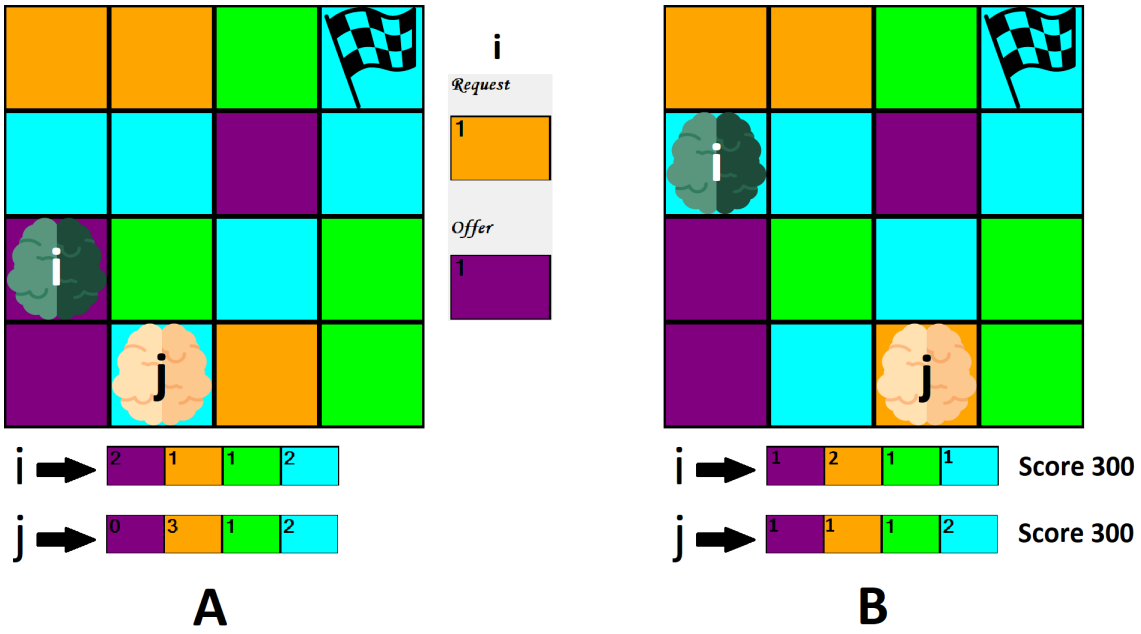


Figure 6.4: Figure A shows the proposal made by agent i, which results in the highest expected value. Figure B shows the game state after agent j accepted the proposal, where after the agents are advanced one step towards the goal

for his opponent and raises his first-order beliefs about the purple chips. This is why agent j predicts a score of 1000 if he would request a purple chip and offer an orange one, whereas trading a green or cyan against orange gives a lower predicted score.

$EV^{(0)}_i$

	1000,0	725,0	400,0	450,0
	1000,0	1000,0	550,0	588,0
	1000,0	1000,0	1000,0	560,0
	1000,0	1000,0	520,0	1000,0

$EV^{(0)}_j$

	450,0	1000,0	450,0	450,0
	450,0	450,0	450,0	450,0
	656,0	862,0	450,0	450,0
	862,0	862,0	862,0	450,0

$b^{(0)}_i$

	0,00	0,50	1,00	1,00
	0,75	0,00	0,75	0,75
	0,80	0,80	0,00	0,80
	0,80	0,80	0,80	0,00

$b^{(0)}_j$

	0,00	1,00	1,00	1,00
	0,38	0,00	0,75	0,75
	0,80	0,80	0,00	0,80
	0,80	0,80	0,80	0,00

Figure 6.5: This figure shows the expected value matrix $EV^{(0)}$ and zero-order belief matrix $b^{(0)}$ for both agents i and j after the first interaction step is completed and the agents are advanced one step towards the goal.

During the second communication phase agent i again selects randomly the highest expected value in his matrix, which happens to be in this case request green and offer purple. The corresponding value in agents' j matrix is 450, which is not the highest value. This agent then responds to the proposal by making a proposal himself, this is shown in Figure 6.6 A. In the same way as his opponent, agent j looks for the highest expected value in his matrix and makes the proposal. Because accepting this proposal will yield agent i a lower expected value, he declines the offer and both agents are advanced towards the goal, as seen from Figure 6.6 B. The expected value and first-order belief matrices after this communication step are shown in Figure 6.7 for both agents. While interacting with each other both agents reveal more information about their chip sets to each other. Beliefs about chip sets influence the values in the zero-order belief matrix, lowering the value if the agent believes that his opponent does not have the needed chip in his possession and vice versa, as explained in 6.1.5.

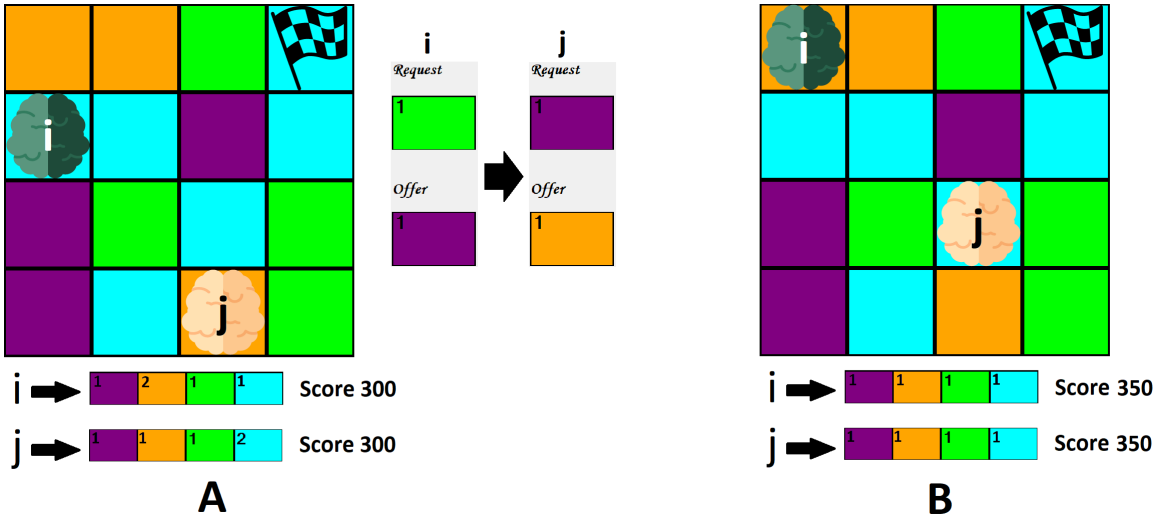


Figure 6.6: Figure A shows the counter proposal made by agent j as a response to the proposal initiated by agent i. Figure B shows the game after agent i declined the counter proposal.

In this example both agents reach the goal simultaneously and get a score of 1000. Because both agents use zero-order ToM it was expected to be a cooperative game because the agents cannot reason about each others mental content such as beliefs. This means that both agents are trying to maximize their own scores, without taking the other agents' preferences into account. This results in continuous negotiation until both agents reach some agreement, not very unlike Multi-Objective Optimization[32]. The next example shows the same game, however it is played by a first-order ToM proposing agent i and zero-order ToM responding agent j.

$EV^{(0)}_i$

	0,0	1000,0	400,0	450,0
	0,0	1000,0	400,0	450,0
	0,0	1000,0	1000,0	560,0
	0,0	1000,0	550,0	1000,0

$EV^{(0)}_j$

	1000,0	0,0	450,0	450,0
	1000,0	0,0	450,0	450,0
	1000,0	0,0	1000,0	588,0
	1000,0	0,0	1000,0	1000,0

$b^{(0)}_i$

	0,00	0,13	1,00	1,00
	1,00	0,00	1,00	1,00
	0,80	0,20	0,00	0,80
	0,75	0,75	0,75	0,00

$b^{(0)}_j$

	0,00	0,50	1,00	1,00
	0,25	0,00	1,00	1,00
	0,75	0,19	0,00	0,75
	0,75	0,75	0,75	0,00

Figure 6.7: Both expected value matrices show that the agents are confident in reaching the goal. The belief matrices after some communication steps show how sure the agents are that a certain proposal will be accepted by the opponent. More interaction with each other increases the accuracy.

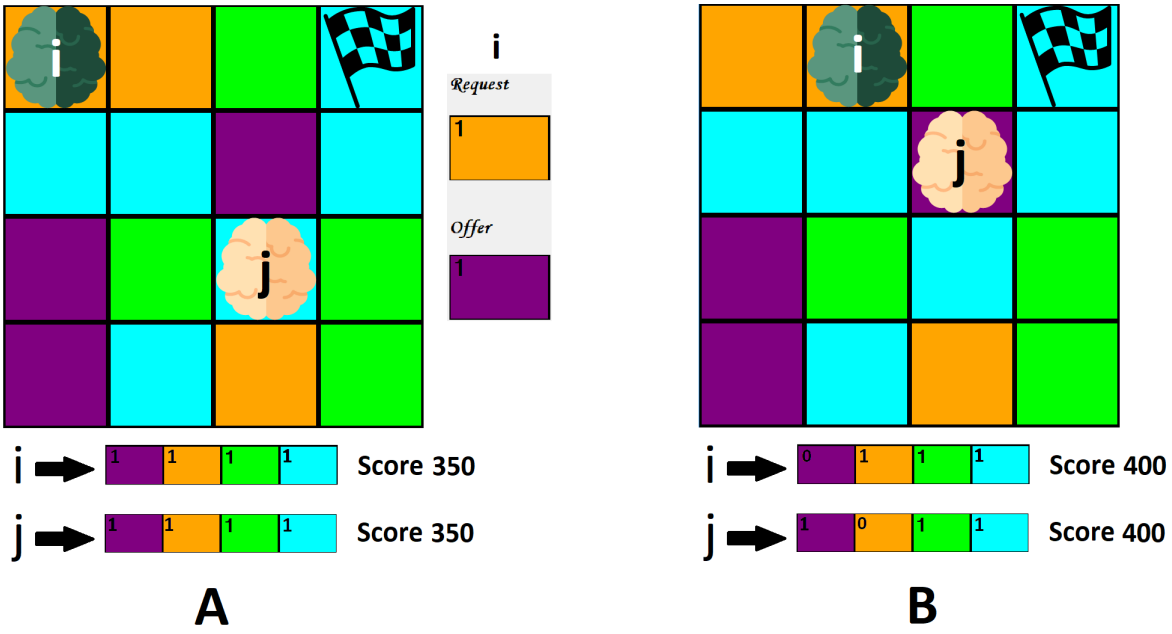


Figure 6.8: Figure A shows the proposal made by agent i. While both agents do not need those additional chips they still agree on the exchange. This is because the chips do not lower their best expected values. Figure B shows the game after agent j accepted the proposal.

Example 2 This example shows the difference in reasoning between a zero-order and first-order ToM agents. The environment for this example is the same as in the previous

example, to be able to see the difference in behaviour more clearly. During the first interaction steps the ToM_1 agent reasons at a zero-order level because he does not have any information about his opponent yet. This means that the first interaction step is similar to the one shown in the previous example. For simplicity the whole game scenario is given in Figure 6.9, where the red numbers indicate the game state at time t.

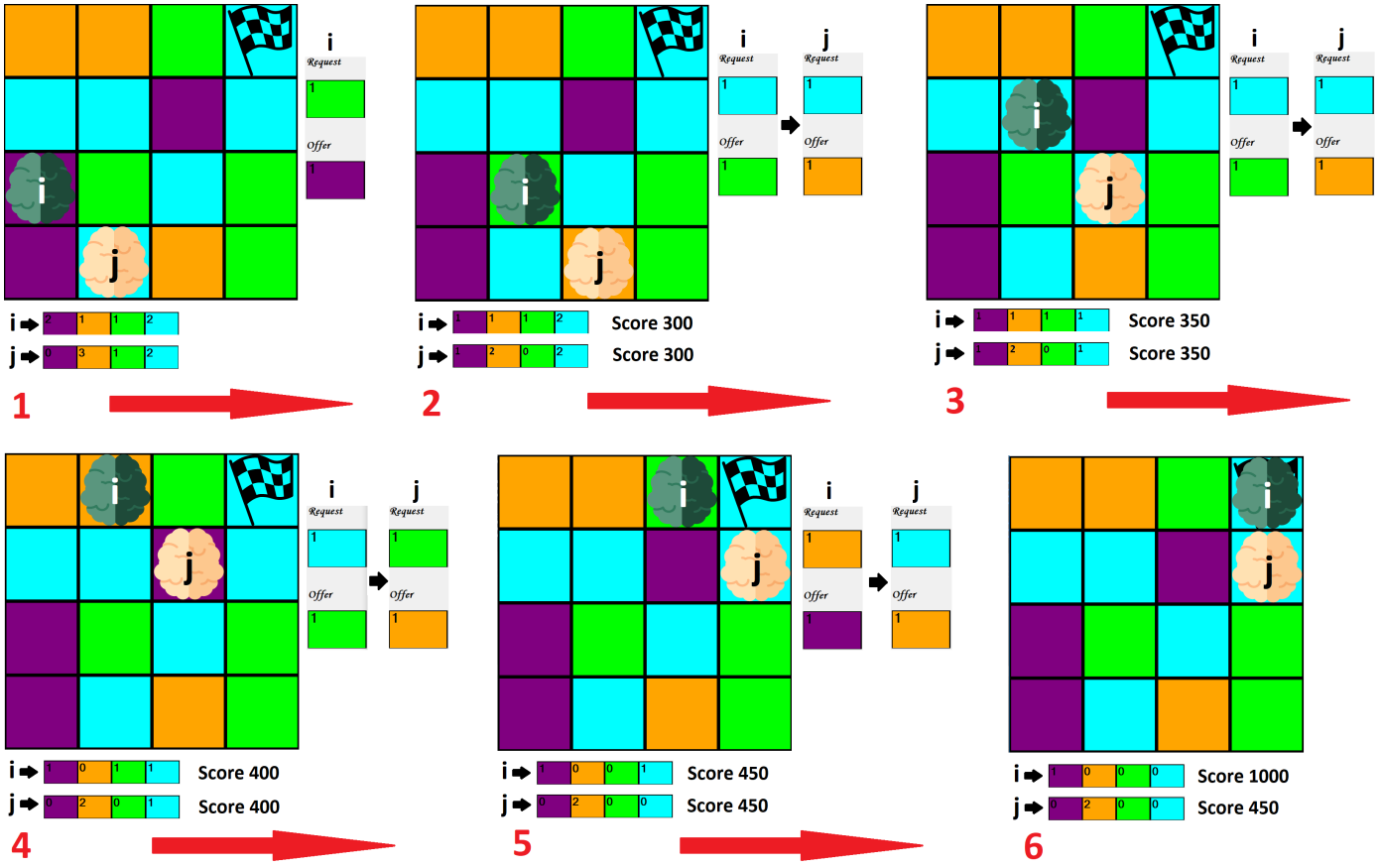


Figure 6.9: The entire game process, where agent i is a ToM_1 agent and j is ToM_0. Agent i only makes an initial proposal, which is accepted by the agent j, after that all counter proposals are declined.

During the first interaction step the ToM_1 agent i forms the same beliefs and expected value matrices as if he were a ToM_0 agent. After second step the agent has some more information about agents' j chip set, namely he has to have at least one orange chip in his chip set. Agent i now sets the likelihoods of chip sets that have zero orange chips to 0.0 in his chip set beliefs. From this observation the agent updates his first-order belief matrix and thereby also his expected value matrix. With the increasing confidence in his first-order model, the agent now computes his first-order expected value matrix. The matrices after the second communication step are given by Figure 6.10.

Note that even though agent i has won the game, deception has not occurred. The reason why the agent won the game is because in this situation he was able to look further into the communication process. After the initial trade agent i only requested the chips

$EV^{(1)}_i$

	998,0	998,0	873,0	680,0
	691,0	998,0	664,0	563,0
	998,0	998,0	998,0	707,0
	998,0	998,0	998,0	998,0

$EV^{(0)}_j$

	450,0	450,0	0,0	400,0
	370,0	450,0	0,0	410,0
	450,0	450,0	0,0	450,0
	375,0	656,0	0,0	450,0

$b^{(1)}_i$

	0,00	0,74	0,37	0,74
	1,00	0,00	1,00	1,00
	0,74	0,74	0,00	0,74
	0,74	0,37	0,37	0,00

$b^{(0)}_j$

	0,00	1,00	1,00	1,00
	0,80	0,00	0,80	0,80
	0,50	1,00	0,00	1,00
	0,75	0,38	0,38	0,00

Figure 6.10: First-order expected value matrix and first-order belief matrix as computed by agent i after communication step 2. Both zero-order matrices of agent j are also given.

that would be declined by agent j, because by then he had enough chips to reach the goal. In cases such as this example, where interaction between agents is minimal, ToM_1 agents are unable to model the other agents' behaviour precisely because of the lack of information about that agents' chip set. This leads to a larger focus on his zero-order beliefs, as the confidence in his first-order model does not exceed 0.5

6.3 Parameters

This section describes the parameters used for running the simulations. The parameters that influence the general setting of the game are kept the same as those that were used in the previous case study:

Board

Dimensions of the board are chosen as 4 x 4 where 4 colours are randomly distributed among each tile. Reason for choosing such dimensions and the amount of colours is again to reduce the computational resources. Because in a repeated game the agents have to make calculations at each round it is expected that this scenario is more computationally expensive than the one shot game.

Agents

In order to investigate the occurrence of deception in agents using Theory of Mind, both agents are allowed to reason on different orders of ToM. The simulations are run with zero-order, first-order and second-order ToM agents, using every combination on the responder and proposer. This enables for investigation of the agents behaviours when presented different orders of ToM opponents.

Because both agents have the same amount of information where some of it is unknown, they need to interact more often with each other in order to form beliefs about that information. This is done by giving the responder the ability to make a counter offer. The agents are allowed to make a series of counter offers until they come to an agreement and are advanced one step towards the goal. Subsequently the negotiation process starts again and the proposer makes his new proposal. Over the course of the game the proposer will make in general more offers than the responder. Research into negotiation shows that the opening bid of a negotiation process is important for the outcome, where the person that makes the initial offer has a disadvantage over the other[33][34]. Keeping that in mind, simulations are also run with the roles of the agents reversed.

Chip sets

The parameters for agents' chip sets are the same as in the previous case study, where the maximum amount of a certain colour in a chip set is 4 as seen in chapter 4.2. The same chip set rules still hold for both agents, where they cannot reach the goal given their initial distributions of chips.

Scoring

Four parameters in the scoring function are defined as follows:

1. *Goal weight*, set to 500 to ensure that both players have incentive to move towards the goal as it increases the score the most.
2. *Distance*, set to -50, the penalty a player receives for each square away from the goal along the Manhattan distance.
3. *Chip weight*, set to 0 to ensure that each player will try to advance towards the goal at each turn. If the chip weight would have been set to a positive number the agents would have no incentive to move towards the goal when lacking the information about opponents chip set and his behaviour. This is because the only way to gain a higher score is by reaching the goal, which cannot be predicted with certainty.
4. *Base score*, set to 500 to prevent from agents having negative scores. Because the agents are playing on a 4 x 4 board, the distance from an agent to the goal cannot exceed 5 steps. This means that the agents start with a minimum score of 250 and upon reaching the goal the agents would receive a total score of 1000.

6.4 Simulation Results

Simulations for this case study were performed using pairs of different orders of Theory of Mind agents. These pairs are as follows:

- Zero-order proposer i against zero-order responder j: i_0 - j_0.
- First-order proposer i against zero-order responder j: i_1 - j_0.
- Zero-order proposer i against first-order responder j: i_0 - j_1.
- First-order proposer i against first-order responder j: i_1 - j_1.

Because the agent that makes the initial offers has a disadvantage by revealing more information about his chip set, the above simulations are also run with the roles reversed as mentioned in section 6.3. This will make the agent i the responder and agent j the proposer.
Similar to the previous case study a 1000 different game boards were generated where the agents played on a total of 10 times for each board. These runs were performed for every scenario, with the same 1000 boards, in order to make relevant comparison of the outcomes. Playing on the same board multiple times sometimes result in different outcomes because the agents make a random decision when choosing between actions that result in equal expected scores. This means that in order to make relevant analysis there need to be as many alternative outcomes as possible. The measures consisted of the scores the agents received, amount of deceptions each agent has made and the amount of cooperative actions. These metrics allow for comparison of agents' performances using different orders of ToM and to show the differences between the roles of the agents. Table 6.4 shows the average scores and standard deviations over all runs for agents using different orders theory of mind and with their roles reversed.

	f^i	f^j	σ_i	σ_j
i_0 - j_0	771.2	781.8	291.6	285.8
i_1 - j_0	766.0	774.5	293.6	287.7
i_0 - j_1	764.1	772.7	293.2	288.7
i_1 - j_1	755.0	764.4	296.0	290.4
Rev i_0 - j_0	781.0	770.3	289.4	288.4
Rev i_1 - j_0	775.8	764.2	291.7	289.8
Rev i_0 - j_1	771.4	762.8	292.1	290.9
Rev i_1 - j_1	766.0	757.1	294.1	292.7

Table 6.4: Average scores f and standard deviations σ agents received when playing a 1000 generated games 10 times for different combinations of orders of Theory of mind. All scenarios use the same 1000 boards for consistency.

The results show that the responding agent j always has a higher score than the proposer i. With the roles reversed the outcomes support the claim, because in that case the agent i gets on average higher scores than agent j. This is because the proposing agent reveals more information about his chip set than his trading partner. Having

more information about the other agents' chip set gives the responder the ability to find better solutions than he otherwise would. Interestingly, even when ToM_1 proposing agent plays against a ToM_0 agent he gets a lower score on average. This is mainly because the ToM_1 proposer is unable to predict the behaviour of his opponent during the first communication as he has no information about the opponents chip set yet, making his confidence in his first-order model 0. This results in the ToM_1 agent using his zero-order beliefs during the start of the game. Because in a lot of cases the agents only need to trade one chip to reach the goal the ToM_1 agent has no good opportunity left to make a better trade when using his first-order beliefs. Additionally, in cases where a ToM_1 agent is able to model the behaviour of his ToM_0 opponent he will make offers that are more beneficial to his opponent, according to his beliefs, in order to prevent him from declining the offer and thereby withdrawing from negotiations. Even though a ToM_0 agent is unable to reason about the intentions and beliefs of his trading partner, when both ToM_0 agents are playing they are often able to increase their scores through negotiation. Surprisingly, Table 6.4 shows that the standard deviation σ_j of agent j is always smaller than that of agent i, even with the roles reversed. This indicates that the scores agent j has obtained do not vary as much as the scores of agent i, which is surprising seen the model of agent j is identical to agent i. The standard deviation values show that dispersion in the set of obtained scores increases with the increase of ToM order. Because agents using first-order ToM are continuously switching between orders based on opponents' behaviour and are able to deceive the opponent more often, the eventual scores vary more. When comparing that increase in standard deviation with the increase of deceptions for each order ToM agent as shown in Table 6.5, a noticeable positive correlation between the two can be seen.

Both agents' scores decrease when at least one of them has a first-order ToM reasoning as opposed to both zero-order ToM agents. This is because a ToM_1 agent tries to find more optimal solutions for himself, increasing the amount of deceptive moves as opposed to cooperative, as seen from Table 6.5. This behaviour is also seen in two ToM_1 agents playing against each other, resulting in lower obtained average scores for both agents than cases where just one of the agents uses first-order ToM reasoning. Both agents' desire to obtain the highest possible score results in a decrease in the overall obtainable score.

There is a clear difference between the amount of cooperative and deceptive actions made by ToM_0 and ToM_1 agents. Table 6.5 shows the average amount of cooperative and deceptive moves agents i and j made for each 1000 games and rounded to the nearest integer.

This table shows that ToM_0 agents are able to cooperate with each other quite well, while keeping a low amount of deceptive moves. The ratio between the cooperative

	i has deceived	j has deceived	cooperations
i_0 - j_0	271	293	765
i_1 - j_0	1026	295	1005
i_0 - j_1	467	951	1049
i_1 - j_1	1185	1054	1172
Rev i_0 - j_0	249	284	772
Rev i_1 - j_0	961	515	1065
Rev i_0 - j_1	283	1125	1073
Rev i_1 - j_1	1102	1210	1289

Table 6.5: Average amount of deceptive and cooperative actions the agents have made. The values are averaged over 10 runs of 1000 games, rounded to the nearest integer.

actions and deceptive actions explains the differences in scores. Having more deceptive actions than cooperative decreases the overall score of the agents. It can be seen as having a desire to obtain the largest piece of pie resulting in a smaller pie to share. Note that because both ToM_0 agents are unable to reason about their opponents' beliefs the amount of deceptive actions in the above table are the result of *unintentional deception*. The amount of deceptive moves increases dramatically with the introduction of a ToM_1 agent, which supports the hypothesis $H3$ as stated in 1.3. However, deceptive behaviour is not beneficial for agents playing in this mixed-motive setting, as seen from their scores. This claim does not agree with the hypothesis $H1$ for this setting, because the results show that cooperative behaviour results in a overall higher score.
ToM_1 agents engage in negotiations more often while trying to increase their confidence in their first-order beliefs, which explains the high amount of deceptive actions. The table shows when two ToM_1 agents play together, they are able to deceive each other roughly equal amount of times. This observation, along with the roughly equal amount of deceptions when two ToM_0 agents play, supports the hypothesis $H5$ for zero-order and first-order ToM agents.

6.5 Hypotheses Test

The significance of the scores and the amount of deceptive moves between the agents i and j is shown in the next subsection by performing parametric tests on the obtained data. Specifically, the tests performed are two sample z-tests assuming unequal variances, because the samples are taken from two different populations. The reason for choosing z-test over t-test is because of the large sample sizes. For these tests the degrees of freedom is set on infinity because the sample size for each population is 10000, making the t table a standard normal distribution z. Additionally, because the information about the population is known, which includes the mean, it is expected that

more accurate results will be obtained using a parametric test over a non-parametric one. The first tests are performed on the set of scores obtained by the agents, followed by the tests on the amount of deceptive moves each ToM agent has made.
Second subsection calculates Pearson correlation coefficients between deceptive and cooperative actions, in order to see how deception correlates with cooperation, for both agents using zero-order and first-order ToM.

6.5.1 Two sample z-tests

The first set of z-tests is performed on the agents' scores, in order to show whether there is significant difference. The null hypothesis H_0 is stated as follows: $H_0 : \mu_i = \mu_j$. Accepting the null hypothesis implies that there is no difference in scores, meaning that the scores could have been obtained by a random chance. Rejecting the null hypothesis implies a significant difference in scores, $H_1 : \mu_i \neq \mu_j$. The significance level for the tests is set to $\alpha = 0.05$. Table 6.6 shows the results of performed z-tests on different orders of ToM agents.

	SE_i	SE_j	*z-cal*	*z-crit*	p	*Decision*
i_0 - j_0	2.92	2.86	-2.599	1.96	0.0094	Reject H_0
i_1 - j_0	2.94	2.88	-2.048	1.96	0.0406	Reject H_0
i_0 - j_1	2.93	2.89	-2.085	1.96	0.0371	Reject H_0
i_1 - j_1	2.96	2.91	-2.279	1.96	0.0227	Reject H_0
Rev i_0 - j_0	2.89	2.88	2.611	1.96	0.009	Reject H_0
Rev i_1 - j_0	2.92	2.90	3.055	1.96	0.0022	Reject H_0
Rev i_0 - j_1	2.92	2.91	2.072	1.96	0.0383	Reject H_0
Rev i_1 - j_1	2.94	2.93	2.133	1.96	0.0329	Reject H_0

Table 6.6: Table showing the results of different z-tests between the average scores obtained by the agents, using significance level $\alpha = 0.05$. SE_i and SE_j are the standard errors for agents i and j respectively, *z-cal* is the calculated z value. For clarification the p value approach is shown as well.

The results show a significant difference in mean scores between agents using different orders of ToM and having different roles, resulting from rejection of the H_0. In order to see the difference in agents' deceptive behaviour, z-tests were performed on the amount of deceptions. Again, the null hypothesis H_0 states that there is no significant difference in the amount of deceptions when two agents are playing against each other, $H_0 : \mu_i = \mu_j$. Rejecting the null hypothesis implies significant difference. The results are shown in Table 6.7 below.

The results show a clear difference in the amount of deceptions done by both agents, which means that different orders of ToM deceive differently. The z values that come

	SE_i	SE_j	*z-cal*	*z-crit*	*p*	*Decision*
i_0 - j_0	0.0047	0.0050	-3.32	1.96	0.0009	Reject H_0
i_1 - j_0	0.0185	0.0055	41.373	1.96	< 0.0001	Reject H_0
i_0 - j_1	0.0060	0.0155	-29.146	1.96	< 0.0001	Reject H_0
i_1 - j_1	0.0176	0.0169	5.337	1.96	< 0.0001	Reject H_0
Rev i_0 - j_0	0.0048	0.0049	-5.04	1.96	< 0.0001	Reject H_0
Rev i_1 - j_0	0.0162	0.0064	25.646	1.96	< 0.0001	Reject H_0
Rev i_0 - j_1	0.0051	0.0183	-44.452	1.96	< 0.0001	Reject H_0
Rev i_1 - j_1	0.0176	0.0174	-4.386	1.96	< 0.0001	Reject H_0

Table 6.7: Table showing the results of different z-tests between the amounts of deceptive behaviours as performed by agents, using significance level $\alpha = 0.05$. SE_i and SE_j are the standard errors for agents i and j respectively, *z-cal* is the calculated z value. For clarification the p value approach is shown as well.

the closest to the critical value are obtained with the results from scenarios where ToM_0 agent i plays against ToM_0 agent j and the reversed roles scenario where ToM_1 agent i plays against ToM_1 agent j. This is because in both scenarios the agents are using the same ToM order reasoning, which results in roughly equal amount of deceptive actions.

6.5.2 Significance Test using Pearson correlation

Pearson correlation is used to show the correlation between deceptive and cooperative actions. This measure and calculation steps are described in-depth in section 5.5.3, and will not be discussed here. In order to find how deception and cooperation correlates in scenarios where different orders ToM agents play Colored Trails with each other, we need to compute Pearson correlation coefficient. If this coefficient is negative then that means that there is negative correlation between deception and cooperation, meaning that increase in deceptive actions implies decrease in cooperative actions. The significance of this correlation is obtained by computing the t-value using the degrees of freedom and the correlation coefficient. The results are shown in Table 6.8, where the amount of cooperations and deceptions is taken as average for each 1000 games. Because the amount of observations here is the averages for each 1000 games, the *df* will be set on $(10 + 10) - 2 = 18$. The t-values in the table show the significance of the correlation, where red values indicate that the correlation is insignificant because it is smaller than critical t-value 2.101 given $df = 18$.

The significant correlations show that in general the deceptive actions performed by ToM_0 agents correlate negatively with the cooperative actions. This means that the agent can either deceive someone through unintentional deception resulting in less cooperative actions or engage in cooperative negotiation resulting in less deceptive behaviour.

	$mean_i$	$mean_j$	$mean_{coop}$	r_{ic}	r_{jc}	t_{ic}	t_{jc}
i_0 - j_0	270.5	293.4	765.3	-0.2772	-0.3585	-0.816	-1.086
i_1 - j_0	1119.4	321.8	1096.3	0.1371	-0.2116	0.392	-0.612
i_0 - j_1	466.5	950.8	1048.9	-0.6833	0.0151	-2.647	0.043
i_1 - j_1	1166.5	1036.5	1119	0.1776	0.3712	0.766	1.696
Rev i_0 - j_0	249.3	283.8	772.4	-0.5550	-0.3464	-1.887	-1.044
Rev i_1 - j_0	960.5	514.5	1064.8	0.4734	-0.6307	2.280	-3.448
Rev i_0 - j_1	282.5	1124.8	1072.8	-0.6006	-0.4290	-2.125	-1.343
Rev i_1 - j_1	1101.7	1210.3	1288.7	-0.0109	0.4636	-0.031	1.480

Table 6.8: Table showing the obtained Pearson correlation coefficients indicated by r_{ic} for correlation between deceptive actions made by agent i and amount of cooperations and the same for agent j, r_{jc}. Using the coefficients the t-values are computed, showing the significance of the correlation, where values in red are not significant enough.

Note however, the significant correlations for ToM_0 agents are obtained only when playing against a ToM_1 agent. This might imply that if a ToM_1 agent detects that he is being deceived, he will not engage in cooperative interactions. Because cooperation is both-sided it will decrease with the increase of deceptive behaviour. Alternatively, if a ToM_1 agent detects cooperative behaviour he will cooperate as well. This will increase the cooperative actions and decrease the ToM_0 agents' deceptive actions.
In cases where both ToM_1 agents play with each other the results do not suggest any significant correlations, meaning that the deceptive behaviour of those agents is not influenced by cooperation.

6.6 Analysis of Results

Results show that agents using zero-order Theory of Mind are able to obtain a higher average score than first-order ToM agents. Even though ToM_1 agents are able to negotiate better, which is seen from the increase in the amount of cooperative actions, they do not have a competitive advantage over ToM_0 agents. When two ToM_0 agents play, the game becomes essentially a multi-objective optimization function[32] where both agents are trying to increase each others score as much as possible, through negotiation. However, with a ToM_1 agent in play, the game is mixed-motive because that agents' behaviour depends on the configuration of the environment and the opponents' behaviour. Being able to reason about opponents' beliefs and intentions gives the ToM_1 agent the ability to predict his behaviour. This however, leaves the agent with a disadvantage, namely the agent can predict which proposals will be declined by his trading partner.

This means that the agent ends up having less options to choose from. Through achieving a cooperative solution the ToM_1 agent gets a lower score than his ToM_0 opponent.

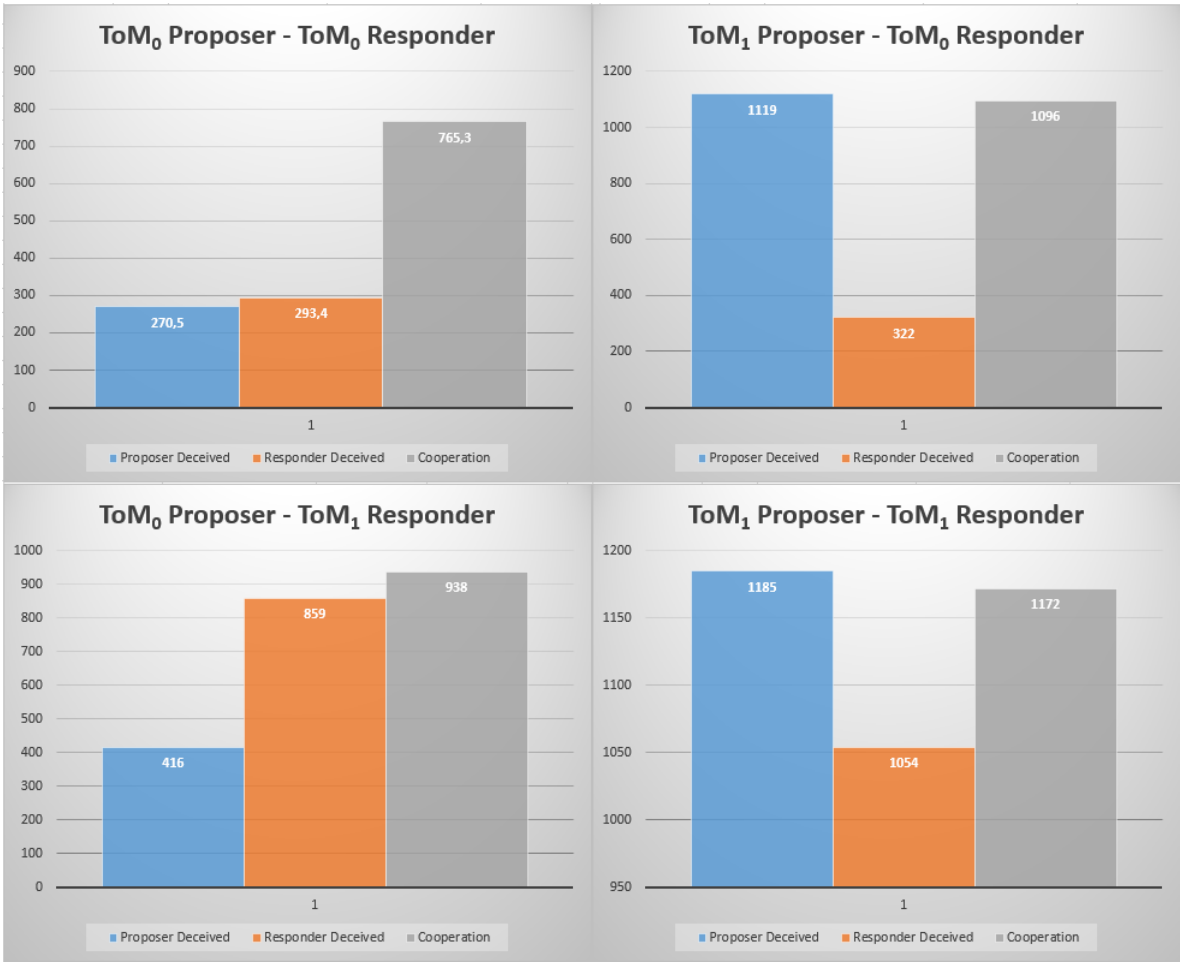


Figure 6.11: Average amount of deceptive and cooperative actions of agents for each 1000 games.

The overall obtained scores are the lowest when two ToM_1 agents negotiate. By trying to predict each others' behaviours and trying to deceive each other at the same time, the agents often fail to reach an agreement when negotiating. By deceiving each other the agents change each others beliefs about the chips they own, resulting in both agents believing that they cannot reach the goal with the chips they own. This leads to both agents terminate negotiations and not being able to reach the goal. This observation rejects the hypothesis $H4$ because ToM_0 agents are often able to reach an agreement and therefore obtain a higher score, unlike ToM_1 agents.

Comparing the scores the responding agent received with the proposer it can clearly be seen that the responder has an advantage, because the overall scores are in his favour. Reversing the roles shows the same results, where the agent that has been changed from proposer to responder gets the highers scores. This is expected, as shown in various

studies on negotiation [33][34] that the opening bid of a negotiation process is important for the outcome.

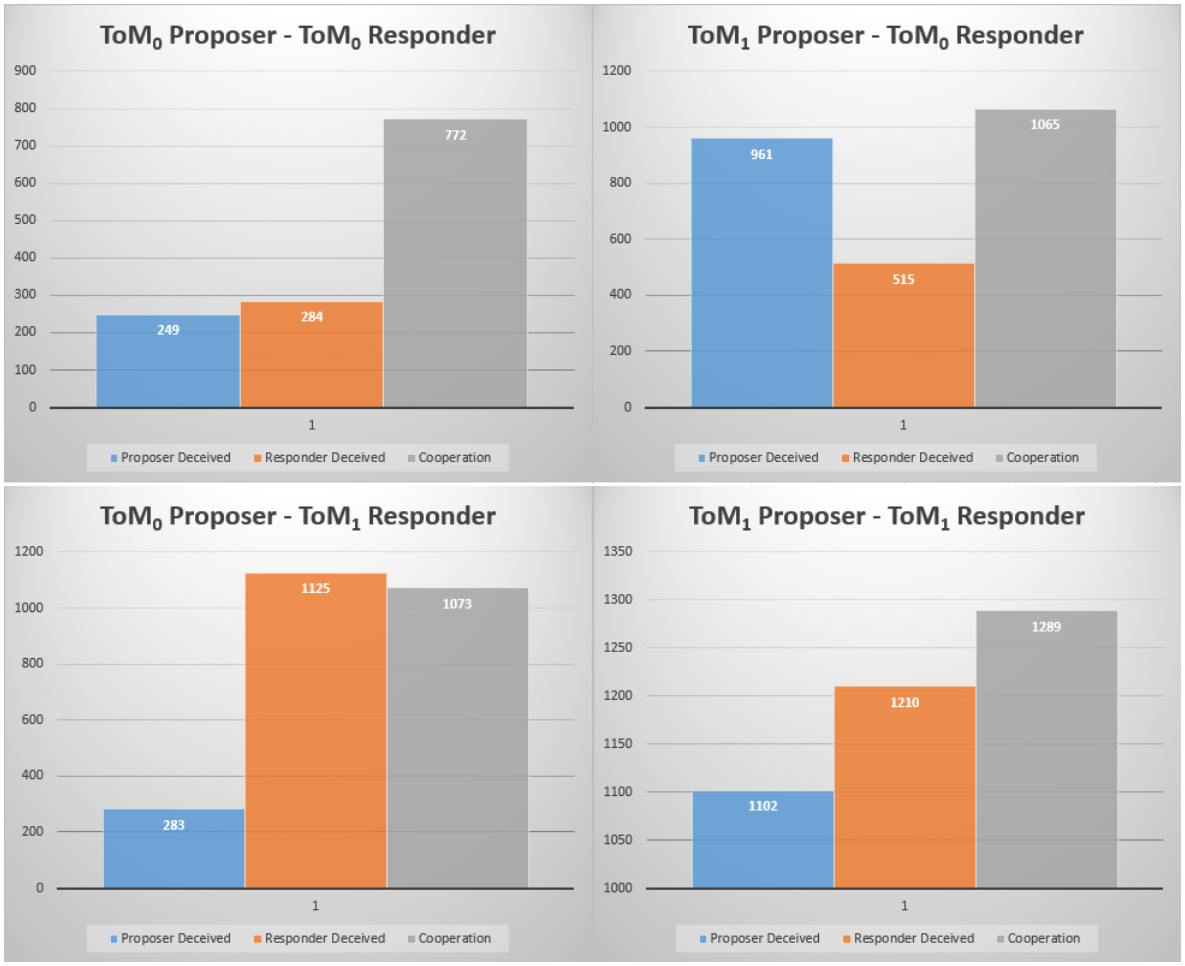


Figure 6.12: Average amount of deceptive and cooperative actions of agents for each 1000 games, with roles reversed.

Figure 6.11 shows the average amount of deceptive and cooperative actions the agents made for each 1000 games. Each chart represents a different scenario where agents' ToM reasoning varies. The results show that ToM_1 agents are able to perform more deceptive actions, which confirms the hypothesis $H3$. Because a ToM_1 agent is able to reason about the observations the opponent makes, he can make a proposal or a response that will change the opponents beliefs about his chip set. This strategy can be used to prevent the opponent from asking a certain chip. The increase in deceptive behaviour is also observed when reversing the roles of the agents. This is shown in Figure 6.12, where ToM_1 agents have a significantly higher amount of deceptions. Noticeably however, equal orders of ToM agents are able to deceive each other roughly equally often, which supports the hypothesis $H5$.

The results of Pearson correlation tests show that agents using zero-order ToM favour cooperative moves over deceptive. This is shown as a negative correlation coefficient. When both ToM_0 agents negotiate, the amount of cooperative actions are higher than deceptive. This behaviour is also observed when reversing the roles of agents. In contrast, the ToM_1 agents receive most of the time a correlation coefficient close to zero indicating that the agents do not favour any strategy. This can be observed when two ToM_1 agents negotiate, where the amount of deceptive actions each agent performs is roughly equal to the amount of cooperative actions. Because the agents play in a mixed-motive environment their behaviour is dependant on the environment set up and the behaviour of the opponent. Figures 6.13 and 6.14 show the correlations between deception and cooperation for agents using zero-order and first-order ToM respectively. It can be seen that ToM_0 agents obtain a negative correlation between deception and cooperation, whereas looking at the trend line for ToM_1 agents it can be concluded that there is no correlation between both values.

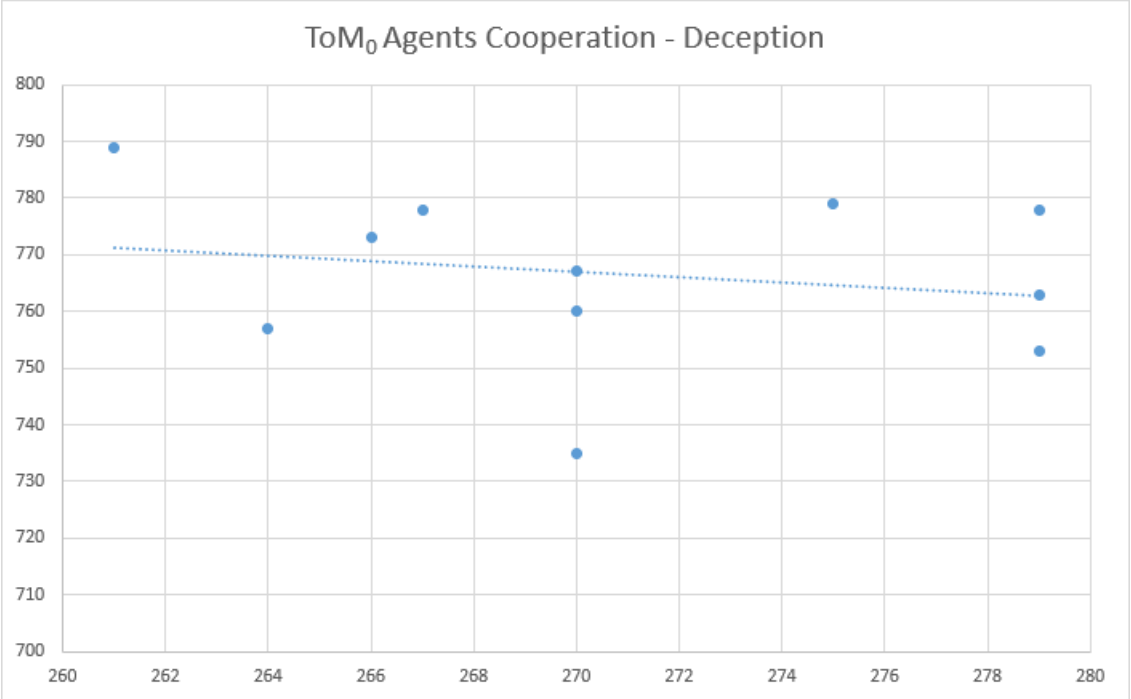


Figure 6.13: Correlation between a sample of deceptive and cooperative actions, as performed by ToM_0 agents. Here the x-axis represents the amount of deceptive actions and y-axis represents the amount of cooperative actions.

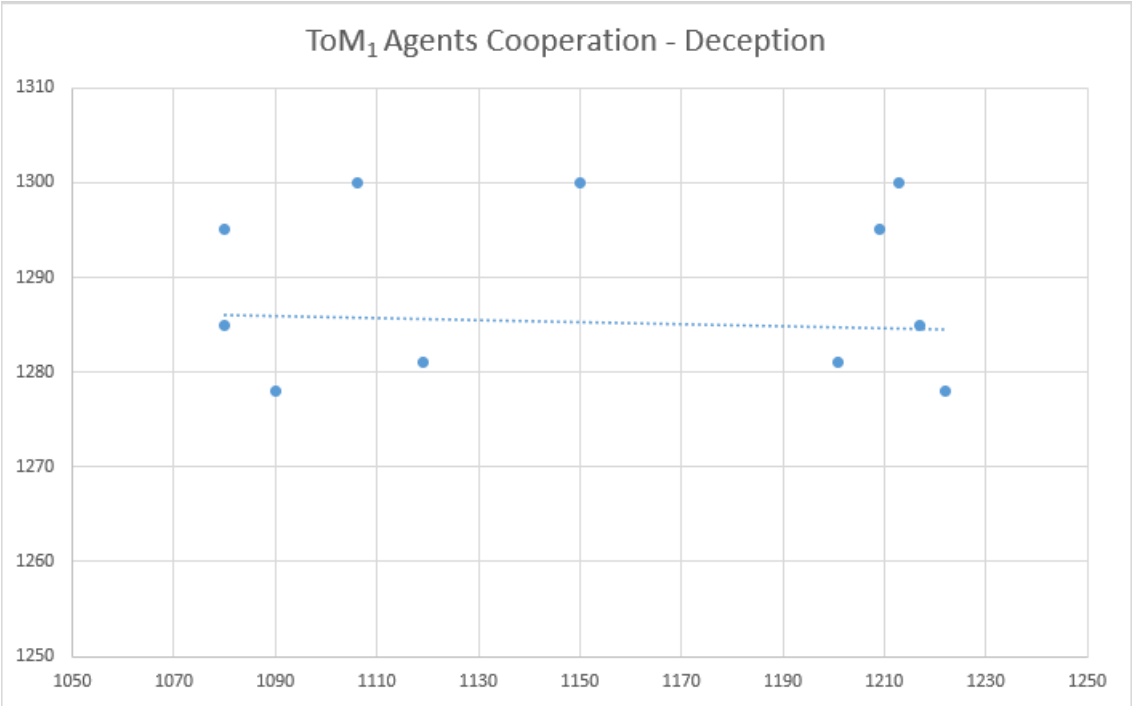


Figure 6.14: Correlation between a sample of deceptive and cooperative actions, as performed by ToM_1 agents. Here the x-axis represents the amount of deceptive actions and y-axis represents the amount of cooperative actions.

Chapter 7

Discussion

The results in this study were gathered using two different methods, in the form of one-shot and repeated games case studies. The environment used to test the hypotheses is a multi-agent mixed-motive interaction framework that has been implemented based on the existing framework called Colored Trails[24][35]. In this environment the agents were able to successfully deceive each other, given the right circumstances. In one-shot games, a first-order theory of mind agent having complete information was able to find deceptive moves most of the time when playing against a zero-order theory of mind agent with incomplete information. This behaviour results in the deceptive agent obtaining much higher average score throughout multiple games. However results show that in same environments given that both agents have the same amount of information, they are able to cooperate with each other quite well. As a result, the average scores of both agents do not fall far below the average score a deceiving agent obtained. This is because in mixed-motive interaction environments cooperative behaviour is sometimes preferred over competitive. The standard deviations of the scores both agents obtained confirm this claim, as shown in Figure 5.13. The values are lower for both agents when they are cooperating more often with each other, whereas the values for the deceiving agent are much higher, indicating a larger spread in obtained scores.

The second case study narrows down onto the behaviour of both agents by allowing frequent interaction during a single game, resulting in a repeated game scenario. Giving both agents the ability to make proposals, interact with each other more often and decide for themselves whether to perform a cooperative or deceptive action shows interesting results. Agents using zero-order theory of mind were able to obtain higher average scores than first-order theory of mind agents. The ToM_1 agents perform better when presented some purely competitive settings as seen in multiple studies[36][37], however this is not the case here. This is because the framework used in this study represents a mixed-motive setting. While in competitive games the agents increase their own score

by decreasing opponents score, in a mixed-motive game the agent might be increasing his score as well as his opponents'. Because ToM_1 agents increase their score by negotiating, a failed negotiation results in a low score. In order to prevent the failure of negotiation the agent has to make proposals that should increase the score of his opponent, hence the ToM_0 agent obtaining a higher score on average when playing against a ToM_1 agent.

Results show that agents using zero-order theory of mind are performing less deceptive actions and more cooperative ones. As a consequence, agents that perform more deceptive actions during the negotiation fail to reach an agreement more often. This direct correlation between the increase in deceptive behaviour versus the decrease in average score can be seen in tables 6.4 and 6.5. However, the results from Table 6.5 show that first-order theory of mind agents are able to find more deceptive actions than agents using zero-order theory of mind. This is explained by the ToM_1 agents' ability to represent the opponents' beliefs and intentions. Being able to predict the opponents' behaviour to a certain degree gives the agent the ability to explore multiple outcomes, including the ones where the opponent fails to model the chip set of that agent correctly. Because of those changed beliefs, the opponent then lowers his best expected values resulting in deception.

Multiple z-tests were performed to compare the scores the agents obtained, as shown in Table 6.6. The results showed that for every combination of theory of mind agents there are significant differences in scores, for a significance level of 0.05 (5%). This means that the results could not have been produced by chance, which confirms the validity of the implementation. The same set of tests were performed on the amount of deceptions produced by both agents, this is shown in Table 6.7. Results from those tests show that the differences between the amount of deceptions for each agent are very significant, where the significance level is $<<$ 0.05 ($<<$ 5%).

The results from Pearson correlation tests show that agents using zero-order theory of mind reasoning have a higher preference to cooperative actions as opposed to deceptive. Table 6.8 shows that ToM_0 agents get a negative Pearson correlation coefficient in all cases, meaning that as the amount of cooperative actions increases the amount of deceptive decreases. This is confirmed when comparing the amount of deceptions as opposed to cooperations when two ToM_0 agents play. Interestingly first-order theory of mind agents do not seem to have a preference for deceptive or cooperative actions, as seen from the Pearson correlation table. While the correlation coefficients are statistically insignificant, nearly every coefficient corresponding to the correlation between deceptive and cooperative actions first-order ToM agents performed lies very close to zero. This indicates that both deceptive and cooperative actions are independent from each other.

7.1 Limitations

It should be mentioned that theory of mind reasoning in orders beyond zero is computationally resource intensive. An agent having such reasoning is using a large set of first-order beliefs besides his zero-order beliefs. It can be seen as the first-order agent trying to consider every possible zero-order opponent. The amount of first-order beliefs depends on the size of the game and the colours on the board. More colours on the board implies more colours in a chip set which results in a larger amount of possible chip sets. This results in a large amount of computation needed in order to perform updates on such set, each time the agent makes a new observation. Additionally, a second-order theory of mind agent would make a set representing all first-order opponents, which in turn each represent all zero-order agents. Such implementation would result in exponential increase in computation time. The process of representing the opponent can be simplified however, reducing the time complexity of the algorithm. This is explained in Chapter 9.

Chapter 8

Conclusion

The research question posed in this thesis was as follows: *Under what conditions can deception emerge in computational agents using Theory of Mind?* To research such conditions a multi-agent interaction environment named Colored Trails was built. This framework was built based on the Colored Trails framework introduced by Grosz et al.[13] in 2004. Here, the environment was set up as a mixed-motive setting, which enables the agents to make a decision between a cooperative and competitive action. Firstly, the behaviour of a first-order Theory of Mind agent was investigated, when playing one-shot games with a zero-order ToM agent. Having complete information about the environment, the first-order ToM agent was able to find deceptive strategies. By using cooperation the agents obtained close to Pareto optimal solutions when both had complete information about the environment. This is because the game becomes a multi-objective optimization problem. However on average the deceptive agent received a higher score than cooperative, confirming the hypothesis $H2$.
The next case study introduced a repeated game scenario, where the agents were able to interact with each other more often. Because this case study focuses on emerging deceptive behaviour, a model of how deception and cooperation are interpreted in this setting was built. Subsequently, simulations of games where zero-order and first-order theory of mind agents play CT were run. Results show that although ToM_1 agents were able to deceive ToM_0 agents more often, which confirms the hypothesis $H3$, the average scores of ToM_0 agents were higher. This observation does not support the assumption made in hypothesis $H1$. Agents using zero-order ToM were able to make deceptive actions as well, which is unintentional deception. Results show that these agents favour cooperative actions over deceptive, whereas first-order ToM agents do not have a preference.
The results from simulations where two ToM_0 agents negotiate showed that these agents perform better than two ToM_1 agents negotiating, because those agents favour cooperation over deception. Because ToM_1 agents deceive and cooperate with each other equally

often, they fail to negotiate in some cases. These results do not confirm the hypothesis $H4$, stating that equal orders of ToM play as if they were ToM_0 agents. However seen the amount of deceptive actions performed by agents of equal orders of ToM shows that the same orders of ToM agents deceive each other roughly equally often, which confirms the hypothesis $H4$.

In this study there were multiple conditions needed for deception to emerge. Using a mixed-motive setting with agents having incomplete information gives the agents the ability to choose their strategy based on opponents behaviour and the configuration of the game. Results have shown that using first-order ToM agents in such environments enables the emerging of intentional deceptive behaviour. Additionally, because this behaviour is very dependent on the configuration of the environment and the behaviour of the opponent, cooperative behaviour happens equally often.

Chapter 9

Further Work

In order to explore the emergence of deceptive behaviour more in-depth, the behaviour of a second-order Theory of Mind agent (ToM_2) needs to be investigated. Within the framework that has been introduced in this study, a model of such agent can be made relatively easily. This can be achieved by expanding the current first-order ToM model, which in turn expands the zero-order ToM model. Computing the expected value using second-order beliefs $b^{(2)}$ and a new confidence variable v^2 that shows the confidence of the agent in his second-order ToM model can be implemented as follows:

$$EV_i^{(2)mn}(b_i^{(0)mn}, b_i^{(1)mn}, b_i^{(2)mn}, C_i^t, v^1, v^2) = \\ v^2 \cdot EV_{i**}^{(2)mn} + (1 - v^2) \cdot (v^1 \cdot EV_{i**}^{(1)mn} + (1 - v^1) \cdot EV_i^{(0)mn})$$

Here the $EV_{i**}^{(2)mn}$ represents the estimated expected value obtained from taking the average of all expected values that the agent has calculated for each possible opponent, in the same way as computing of $EV_{i**}^{(1)mn}$ happens as explained in section 6.1.10.1. Computing and updating such values at every observation is very computationally expensive, as mentioned in the limitations section 7.1. However the computational strain can be decreased. Such second-order ToM agent would try to represent all possible ToM_1 opponents as a large set which in turn represent all possible ToM_0 agents as a large set. The set of all possible ToM_0 agents does not need to be modelled by every ToM_1 model as created by ToM_2 agent, because one set would already capture every possible ToM_0 agent. So computing and updating of such set would happen only once, reducing the need for computational resources greatly.

In the second case study the agents are able to negotiate with each other until they have reached an agreement or fail to negotiate successfully. Having a constraint such as a negotiation penalty, which can be seen as a cost that the agents have to pay every negotiation turn, would show a better distinction in performance between ToM_0 and ToM_1 agents. This penalty would force the agents to withdraw from negotiations if they believed that the cost of negotiation outweighs the predicted score. Because a ToM_1 agent would be able to predict such outcome, it is expected that his behaviour would change accordingly. Instead of engaging in continuous negotiation, such agent would choose his proposals is such way that he expects his opponent to not withdraw from negotiations. This means that a ToM_1 agent would offer chips in such way that it increases the opponents score by a large amount, or alternatively find better deceptive strategies using his beliefs that represent the opponents' tendency to withdraw from negotiations.

The implementation of a multi-agent interaction framework as described in this study allows for investigation of interaction between agents in various other contexts. Because this framework was built with the idea of being extensible to other research into multi-agent interaction, this allows for various other research fields. A reason for doing this is because the original version of Colored Trails is not available any more, as a lot of implementation techniques are depricated. A consideration of using this framework in the research that focuses on interaction between multiple agents is made by a post doc Mor Vered along with Liz Sonenberg and Tim Miller from University of Melbourne. This framework is open source for anyone that wishes to perform research into multi-agent interaction and can be found at `https://git.science.uu.nl/P.Homayun/DeceptiveCT.git`.

Appendix A

Classes in the Framework

The following paragraphs will explain briefly the top-level classes and their functions within the framework.

GameCreator

This class allows for creation of different games, which can be either one-shot game or repeated game. This class can switch between the GUI representation of the game to a console window, which is used for running multiple simulations consecutively. Generating the board, players and chip sets happen in this class. Here the set up of different orders of ToM agents happens as well.

MakeActions

This is a static class that specializes in performing different actions such as computation of expected values, likelihoods, enumeration of chip sets and trading. Reason for being a static class is because there is no initialization needed for this class to perform different actions.

GameWindow

This class implements the front-end behaviour such as drawing the state of the game on the screen as a form object. To do that it needs to interact with different classes such as Board, Player and ShortestPaths through initialization methods. By doing so it gets the information needed to draw the configuration of the board on a bitmap, the representation of the players as an SVG image on that board and show the shortest paths towards the goal.

Board

Board class contains the implementation of different aspects of the game board. This includes a list of colours that are being played with in the current scenario. Further the board consists of a 2D array of Square classes that implement a real and a false colour and whether its a goal or not. Using a 2D array gives us a constant time to lookup a specific square. Board class has a variable amount of rows and columns that can be accessed as read-only. The goal piece can be either accessed or set.

Player

Player class is an abstract class in order to make the code more extensible. Through this two derived classes are implemented, namely Proposer and Responder. This class has variables Point and Position, where the first is a position representation in pixels on the

game board when being drawn and the second is the actual position on the board class. A few Lists of best paths and chipsets are available to the players that can be used to make certain choices in order to increase the score. Players have also a ChipSet which is the amount of chips in their possession, this class will be discussed below. Further, every player has a unique ID, which can be a name.

ShortestPaths

This is a static class that implements different methods of getting the shortest paths towards the goal. The first method just computes a specified amount of shortest paths toward the goal without looking at the chips that are available to the player. The second method computes the shortest paths given a combination of chips a player owns and his opponent. However, an empty chip set for opponent can be inserted in order to compute shortest paths towards the goal given only the chips the player owns. FastShortestPaths method computes a path by simply adding or subtracting the squares from the players position until the goal is reached.

Path

Path class stores a known path as a HashSet of Position classes. Position class implements simply the row and column value and can perform calculations on neighbouring positions. Further, Path class can perform different actions such as getting the required chips for a certain path and check whether a position is contained within a path.

ChipSet

Chipset class implements a HashSet of Tuples of Colour and Integer. The integer corresponds to the amount of chips of a certain colour a player owns. This class can perform different actions concerning the chips such as retrieving the colours, setting number of chips given a certain colour, adding and removing of chips, comparing chipsets and adding two chipsets together. The latter two methods are important when calculating a shortest path that is needed through trading with the opponent.

PathWithChips

This class simply implements the combination of the above two classes. It stores the path with the chipset that is needed to take the path. This class is used by the ShortestPaths class, where its being placed in a queue. By doing this the amount of calculations reduces which in turn increases the performance.

TradeWindow

Similar to GameWindow class, this class is also a front-end form object. This class is initialized when a player decides to trade the chips. It draws the chips offered and the chips requested by the player. If the player is a Human, he can accept or deny the trade by pressing the corresponding button.

Appendix B

Shortest Paths Algorithm

This algorithm uses the combined chip sets of both players in order to find the best possible paths towards the goal. Start position of the player is added to the priority queue. From this queue the smallest element is taken. The elements in the priority queue are ordered based on the chip weights. Then each neighbouring tile is examined. If the neighbours' colour is present in the chip set then that tile is added to the current path. Subsequently this path is put into the priority queue, along with the chips left after removing the colour of the added tile. This is done for each neighbour, which results in multiple sub-paths in the queue. The process starting from taking the minimum value from the queue is repeated until a specified amount of paths is found or the priority queue is empty.

Algorithm 1 Shortest Paths Algorithm

```
1:  combinedChips ← Combine(myChips, opponentChips)
2:  path.Add(start)
3:  PriorityQueue.Add(path, combinedChips)
4:  while NotEmpty(PriorityQueue) and counter < pathsFound do
5:      pathWithChips ← PriorityQueue.DeleteMin()
6:      if pathWithChips.Last() == goal then
7:          paths.Add(pathWithChips)
8:          counter = counter + 1
9:      else
10:         for all neighbours do
11:             if !pathWithChips.Contains(neighbour) then
12:                 newPathWithChips ← pathWithChips
13:                 newPathWithChips.Add(neighbour)
14:                 newPathWithChips.RemoveColor(neighbour.Color)
15:                 PriorityQueue.Add(newPathWithChips)
```

Bibliography

[1] J. Dias, R. Aylett, A. Paiva, and H. Reis. The great deceivers: Virtual agents and believable lies. *CogSci*, 2013.

[2] W. Smith, F. Dignum, and L. Sonenberg. The construction of impossibility: a logic-based analysis of conjuring tricks. *Frontiers in psychology*, (7):748, 2016.

[3] N. Abe, M. Suzuki, E. Mori, M. Itoh, and T. Fujii. Deceiving others: distinct neural responses of the prefrontal cortex and amygdala in simple fabrication and deception with social interactions. *Journal of Cognitive Neuroscience*, 2(19):287–29, 2007.

[4] D. Premack and G. Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(4):515–526, 1978.

[5] J. Sobel. Lying and deception in games. *University of California-San Diego, Working Paper*, 2016.

[6] D. B. Buller and J. K. Burgoon. Interpersonal deception theory. *Communication theory*, 6(3):203–242, 1996.

[7] S Ericsson. The ways we lie. *Patterns for college writing: A rhetorical reader and guide.*, (12):477–478, 2011.

[8] P. H. Kriss, R. Nagel, and R. A. Weber. Implicit vs. explicit deception in ultimatum games with incomplete information. *Journal of Economic Behavior Organization*, (93):337–346, 2013.

[9] J. P. Hespanha, Y. S. Ateskan, and H. Kizilocak. Deception in non-cooperative games with partial information. *Proceedings of the 2nd DARPA-JFACC Symposium on Advances in Enterprise Control*, pages 1–9, July 2000.

[10] W. Yoshida, R. J. Dolan, and K. J. Friston. Game theory of mind. *PLoS computational biology*, 12(4), 2008.

[11] H. A. Simon. Theories of bounded rationality. *Decision and organization*, 1(1): 161–176, 1972.

[12] W. B. Arthur. Inductive reasoning and bounded rationality. *The American economic review*, 2(84):406–411, 1994.

[13] B. J. Grosz, S. Kraus, S. Talman, B. Stossel, and M. Havlin. The influence of social dependencies on decision-making: Initial investigations with a new game. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, 2:782–789, July 2004.

[14] A. J. Bishara and J. B Hittner. Testing the significance of a correlation with nonnormal data: comparison of pearson, spearman, transformation, and resampling approaches. *Psychological methods*, 3(17):299, 2012.

[15] G. Zlotkin and J. S. Rosenschein. Incomplete information and deception in multi-agent negotiation. *IJCAI*, 91:225–231, August 1991.

[16] F de Rosis, V Carofiglio, G. Grassano, and C. Castelfranchi. Can computers deliberately deceive? a simulation tool and its application to turing's imitation game. *Computational Intelligence*, 3(19):235–263, 2003.

[17] R. E. Neapolitan. Learning bayesian networks. *Upper Saddle River, NJ: Pearson Prentice Hall.*, 38, 2004.

[18] C. Castelfranchi, R. Falcone, and F. De Rosis. Deceiving in golem: How to strategically pilfer help. *Autonomous Agent98: Working notes of the Workshop on Deception, Fraud and Trust in Agent Societies.*, 1998.

[19] D. V. Pynadath and S. C. Marsella. Psychsim: Modeling theory of mind with decision-theoretic agents. *IJCAI*, 5:1181–1186, July 2005.

[20] C. F. Camerer, T. H. Ho, and J. K. Chong. A cognitive hierarchy model of games. *The Quarterly Journal of Economics*, 3(119):861–898, 2004.

[21] M. Bacharach and D. O. Stahl. Variable-frame level-n theory. *Games and Economic Behavior*, 2(32):220–246, 2000.

[22] C. Camerer and T. Hua Ho. Experienceweighted attraction learning in normal form games. *Econometrica*, 4(67):827–874, 1999.

[23] Y. A. Gal, B. Grosz, Pfeffer A. Kraus, S., and S Shieber. Agent decision-making in open mixed networks. *Artificial In-telligence*, 174(18):1460–1480, 2010.

[24] Y. A. Gal, B. J. Grosz, S. Kraus, and S. Pfeffer, A. abd Shieber. Colored trails: a formalism for investigating decision-making in strategic environments. *Proceedings of the 2005 IJCAI workshop on reasoning, representation, and learning in computer games*, pages 25–30, July 2005.

[25] A. Whiten and R. W. Byrne. Machiavellian intelligence ii: Extensions and evaluations. *Cambridge University Press*, 2, 1997.

[26] H. de Weerd, R. Verbrugge, and B. Verheij. Negotiating with other minds: the role of recursive theory of mind in negotiation with incomplete information. *Autonomous Agents and Multi-Agent Systems*, 31(2):250–287, 2017.

[27] H. de Weerd, R. Verbrugge, and B. Verheij. Agent-based models for higher-order theory of mind. *Advances in Social Simulation*, pages 213–224, 2014.

[28] F. Ducho, A. Babinec, M. Kajan, P. Beo, M. Florek, T. Fico, and L. Juriica. Path planning with modified a star algorithm for a mobile robot. *Procedia Engineering*, (96):59–69, 2014.

[29] P. van Emde Boas, R. Kaas, and E. Zijlstra. Design and implementation of an efficient priority queue. *Mathematical systems theory*, 1(10):99–127, 1976.

[30] R. E. Korf. Depth-first iterative-deepening: An optimal admissible tree search. *Artificial intelligence*, 1(27):97–109, 1985.

[31] M. Allwood. The satterthwaite formula for degrees of freedom in the two-sample t-test. *The College Board*, 2008.

[32] K. Deb. Multi-objective optimization. *Search methodologies*, pages 273–316, 2005.

[33] M. J. Cotter and J. A. Henley Jr. First-offer disadvantage in zero-sum game negotiation outcomes. *Journal of Business-to-Business Marketing*, 1(15):25–44, 2008.

[34] D. Van Poucke and M. Buelens. Predicting the outcome of a two-party price negotiation: Contribution of reservation price, aspiration price and opening offer. *Journal of Economic Psychology*, 1(23):67–76, 2002.

[35] S. G. Ficici, Y. A. Gal, and A. Pfeffer. The colored trails framework: Modelling human negotiation in strategic games. *MURI Meeting on Computational Models for Belief Revision, Group Decisions, and Cultural Shifts*, January 2006.

[36] J. R. Wright and K. Leyton-Brown. Beyond equilibrium: predicting human behaviour in normal form games. *AAAI*, 2010.

[37] H. De Weerd, R. Verbrugge, and B. Verheij. How much does it help to know what she knows you know? an agent-based simulation study. *Artificial Intelligence*, (199):67–92, 2013.