An Epistemic Approach to Analyzing and Modeling Knowledge in Blufpoker

Ivar Mak (S2506580) Oscar de Vries (S2713748) Joost Warmerdam (S2591367)
Logical Aspects of Multi-Agent Systems - University of Groningen

July 6, 2020

1 Introduction

Blufpoker is a dice game which contains several interesting aspects with regards to knowledge, beliefs and bluffing. In this project, we implement Blufpoker and simulate agents playing against each other, using different strategies. This game is implemented as a Multi-Agent System where agents interact with each other and knowledge about other agents is relevant for decision making. In order to reason about what an agent knows, a formalization of the knowledge is made using epistemic logic. In the game, dice rolls are hidden to other players, but unrolled dice are to be revealed. Logically, this functions as a public announcement, leading to common knowledge, which is where the reasoning aspects come into play. In blufpoker, the aspect of bluffing comes into play when a player rolls a bad turn, in which they will have to announce a possible world different from the actual world. Given the knowledge, probabilities of bids can be determined by a knowledge agent.

Furthermore, a model is proposed about the knowledge and reasoning of an agent in Blufpoker. Using this model the knowledge utilized in the game is formalized in a Kripke model. In the simulation, the knowledge agent will reason on the basis of possible worlds that can be reached from different in-game situations, given the common knowledge. The possible worlds will be used for the decision making process of the knowledge-based agent.

We aim to measure the performance of an agent playing according to knowledge, opposed to that of more 'naive' strategies, in which agents play according to more simple rules. Next to this, we aim to visualize the thought process of the knowledge agent in different stages of the game. This will be done by a graphical user interface, which gives a clear overview of how the game is played.

Before addressing the technical aspects of this research, first the description of the game and it's rules will be given.

1.1 Game Description

In this section the game of blufpoker and it's rules will be described. Every round the players take turns rolling three dice, which are concealed under a cup (i.e. other players do not see the player's dice). The aim of the game is to roll a number that is higher than the number that was passed on by the previous player, in order to not lose the round. The dice together depict a number in the hundreds. The highest ranking die translates to the hundred, the second to the ten, and the lowest ranking die corresponds to the single digit. For example, in a situation where a 5, 3, and 2 are rolled the corresponding number is 532. The exceptions are the "pokers" which are triplets (i.e. 111, 222, 333, etc). These are the highest possible rolls: 111 is higher than 665, and 666 (pronounced "poker 6") is the highest possible score. Also, whenever a poker is rolled, this is handled differently than the other rolls, which will be described later in this section. Table 1 describes all possible combinations, ranked in descending order from highest to lowest.

666	664	652	632	552	532	441	332
555	663	651	631	551	531	433	331
444	662	644	622	544	522	432	322
333	661	643	621	543	521	431	321
222	655	642	611	542	511	422	311
111	654	641	554	541	443	421	221
665	653	633	553	533	442	411	211

Table 1: This Table shows the set of possible dice combinations in the game of *Blufpoker*, ranked from top left to bottom right, descending per column.

When the round starts, an agent rolls the dice and announces a claim of what number he has rolled. Apart from the first roll, this announcement has to be higher than the previous announcement. Now the next players' turn starts. It is up to this player to determine whether his adversary is telling the truth. If he believes the truth was told, the player rolls the dice to reach a higher number yet again. In the case he does not believe the previous player, he can call the bluff and lift the cup. If the number that was announced is actually on the table player 2 loses the round, if the number is not there; player 1 loses the round. Apart from the first roll, the agent can roll the dice one at a time and decide to not roll remaining dice. The dice that are not rolled, are shown and become common knowledge. Obviously, there is a possibility that he did not reach a number that is high enough. In this case the player needs to bluff, by announcing a number that he did not roll. Since some of the dice are concealed, he can say whatever number he deems suitable, and pass on the dice to the next player. When a player is caught bluffing, that player loses the round and receives one penalty point.

The first player to receive five penalty points loses the game. Therefore there will always be one loser and multiple winners. The number of penalty points chosen to finish can resemble a word such as "HORSE" or "LOSER", receiving a letter for each loss.

1.1.1 Poker phase

The poker phase is slightly different than the usual rolling phase. This section describes how that phase is executed. Whenever some player announces a poker and the next player believes this (regardless of whether there truly is a poker), the game goes into the poker phase. In this phase, the player (that believed the poker) has three chances to roll an equal or higher poker than the bid it believed, which happens openly (not under the cup). In these three rolls, the player can leave aside any dice he wants, or roll them all again, as long as it are three rolls in total.

If the player then does not manage to roll an equal or higher poker, he gets a penalty point. If he manages to roll an equal poker, no player gets a penalty point. Finally if he rolls a higher poker, the player who initially called the poker gets a penalty point.

1.2 Simplifications

To make programming and simulating the game more manageable as well as distributing the knowledge better, we have implemented several simplifications to the rules of the original game.

- *Only dice combination bids are possible.*
  In the original game players can bid any number they want, even if it is not an actual possible dice roll (e.g. '500'). This aspect does not add much to the game, but does make modelling the game much more complex, which is why we decided to omit this option.

- *Players play to win.*
  In the original game a player might decide to underbid 'tactically', in order to complicate the turn of a specific opponent (this goes hand in hand with the following point). In order to simplify the reasoning and decision making process of our agents, we decided that agents will bid truthfully (if possible).

- *Players always roll.*
  In the original game a player can decide not to roll the dice, bid a higher number than his predecessor, and pass on the cup without looking at the dice. This might be useful in certain scenarios, but in order to simplify our simulation we decided to omit this option.

- *Unrolled dice are placed outside the cup.*
  In the original game players can decide which dice they keep hidden by leaving them under the cup, and which dice to show openly after each turn. We have decided to adhere a specific rule for this: dice that are not rolled during a players turn will be placed outside the cup. This means all players are able to see these dice, effectively making them 'common knowledge'. We chose this so that the chance of knowledge being distributed is higher during the rounds, which gives our knowledge agent more information to base its strategy on.

1.3 Framework

First the implementation of the game is addressed, along with the strategies that are implemented. These strategies will include superficial (or naive) strategies and one complex strategy based on the epistemic knowledge model. Then the implementation of the epistemic model is shown as well as a description of the strategy that is modelled based on this epistemic model. The knowledge-based strategy is then used to play games in order to yield results about its behavior versus more naive, superficial strategies. The model will give insight in how well the proposed knowledge-based strategy performs, based on ratios of how often this agent does not lose.

2 Methods

The Blufpoker simulation and the epistemic model are made in Python[1]. The code is built using a number of classes:

- **Cup**, in which the 3 dice are maintained throughout the game and from which dice can be rolled.
- **Player**, in which the strategies, penalty points and knowledge of the agents are maintained.
- **GUI**, which handles the initialization and main functions of the graphical user interface.

[1]Python Software Foundation. Python Language Reference, version 3.6. Available at http://www.python.org

- **Game**, in which the main function in which the game is played. This is done using a state machine which keeps track of the state of the game. From each state, a number of functions can be executed, which regulate the deeper game aspects. For each phase, where possible, the phase is played according to the strategy of the player of the current turn.
- **Main**, from here the game can be called. This can be both in a way in which the game is visualized turn-wise using the GUI, but also in a way suitable for testing, in which nothing is printed or visualized. Here the agents can be set to play a number of consecutive games, such that performance of strategies can be tested.

The game runs in the terminal and is visualized using the User Interface library *Tkinter* [2]. The interface consists of a central panel showing progress of the game, with depicted dice and players, as well as the current bid. Furthermore on the bottom a text box provides game info which is used to clarify phases of the game, as well as the decision making process. To give insight in the knowledge-based strategy, the program contains a panel on the right-hand side which includes the common knowledge, and a graphical depiction of the deductions made by the knowledge agent in terms of possible worlds and how they relate to the current bid. Figure 4 in the Appendix shows a screenshot of the Graphical User Interface that was created.

2.1 Program

The program simulates three agents playing against each other. Different strategies can be executed by the agents, which are applicable to each of the three phases in a turn. One of these agents will use a knowledge-base for its decision making. This reasoning is described in Section 2.2. The other strategies are less complex and regarded as *naive strategies*. The simulation performs the game in a step-wise manner and assigns penalty letters to the losing agent after each round. The simulation will record, which player has lost and how often the knowledge-based agent wins against the naive strategies. This simulation is performed for 10000 runs to create data measuring the performance of the knowledge-based agent.

2.1.1 Strategies

Before the strategies are defined, a single turn is divided into three phases in which the strategies will differ:

1. *Calling phase*: The agent decides whether the previous agent is truthful about its announcement of the rolled dice.
2. *Rolling phase*: The agent rolls the dice.
3. *Bidding phase*: The agent forms a bid and announces this for others to decide to believe it or not.

Strategies influence an agents decision making in the Calling phase and the bidding phase. The strategies of the other 'naive' agents will differ in three manners: (1) deciding whether to believe the previous player's bid or not, (2) deciding which dice they roll and (3) deciding how they

[2] see https://realpython.com/python-gui-tkinter/

will bid (i.e. bid truthfully or not, and how to bluff). Eventually, these strategies will be used to measure how well the knowledge-based strategy performs in comparison to these different strategies.

An overview of the strategies of the naive agents that will be used in the games against the knowledge-based agent is the following:

1. Calling Phase:
    - Believe a bid with a certain probability
    - Believe any bid
    - Believe no bid at all
2. Rolling Phase
    - Roll one of the dice, randomly chosen
    - Roll the n lowest dice, in which n is randomly determined
    - Roll only the lowest die
    - Greedy: Roll any die that is lower than 6
3. Bidding Phase:
    - Bid truthfully, if possible. Otherwise bid a number x above the previous value, where x is randomly determined in a small range
    - Always overbid previous bid: Bid a number x above the previous value, where x is randomly determined in a small range

A number of combinations of these strategies will be used to define naive agents. Following, testing different configurations of such naive agents in combination with a knowledge-based agent will be used for measuring the performance of the knowledge strategy.

2.1.2 Exceptions: First turn and poker phase

Since both the first turn and the poker phase are different than the aforementioned phases, yet do not allow for much variation, it was decided to keep these phases equally for each of the agents (including the knowledge agent):

- In the first turn, an agent will simply roll the dice and truthfully announce what he has rolled.
- In the poker phase, since the dice are rolled openly, it seemed like any player should apply the same strategy, which is quite straightforward. In this phase, a player will first roll the dice, and when this does not lead to a equal or higher poker in the first roll, the player will execute the following steps until he has no rolls remaining, or does reach a poker:
    1. First check if there are two dice with the same value that can form a higher or equal poker. If so, keep these and roll the other dice, otherwise execute step 2.
    2. Keep the highest die, given that this can form a higher or equal poker, and roll the remaining two in order to get that poker.
    3. If none of the dice can lead to a higher or equal poker, roll all the dice.

2.2 Epistemic Model & Relation to Epistemic Logic

The epistemic model that is proposed for modelling the knowledge for the different agents consists of a Kripke model. A Kripke model (Meyer & Hoek, 1995) is defined as a tuple $\langle S, \pi, R_1, ..., R_m \rangle$ where:

1. S is a non-empty set of states and m is an agent in the set of agents $\boldsymbol{A}$.
2. $\pi : S \rightarrow (\boldsymbol{P} \rightarrow \{\boldsymbol{t}, \boldsymbol{f}\})$ is the truth assignment to the propositional atoms per state.
3. $R_i \subseteq S \times S(i = 1, ..., m)$ are the accessibility relations.

Every world is defined as a possible dice roll, each containing a specific dice combination. To formalize, we define a formula as (x, y, z) with $0 < x \leq y \leq z \leq 6$ which represent which dice are rolled. Here, x represents the hundreds, y represents the tens, and z represents the single values. The truth assignment is not used explicitly in the code. Rather, the truth assignment is defined as follows: $\pi(w_x)((4, 2, 1) = \boldsymbol{t}$ if and only if the world contains the dice combination $(4, 2, 1)$. All other formulas are therefore false in this world. So a world w_1 can contain a formula $(4, 2, 1)$ where the current dice roll is 421.

In this model, the set of agents consists of the players, i.e. $\boldsymbol{A} = \{1, 2, 3\}$ when there are three players. The set of states S is defined by all 56 possible dice rolls, shown in Figure 1.
The accessibility relations are defined as follows: a world w_i can be accessed from world w_j by any agent, if that agent considers possible that when rolling a die that is allowed to be rolled, makes possible that the combination in w_i is rolled. For example, if agent x knows that the current world contains $(3, 2, 1)$, then a world with $d_{4,3,2}$ is possible, as it can be rolled by re-rolling the singular die, namely the die with 1. The complete model is not ideal to show in this report, rather the accessible worlds are identified in the GUI of the simulation. Figure 1 shows the accessibility of a small part of the model in a situation that given a dice combination, a single die is rolled. The Figure shows that there are six possible worlds, which correspond to all possible values on the die that is rolled. If two dice are rolled, 21 worlds accessible. In the case of rolling all the dice, all 56 worlds are accessible.

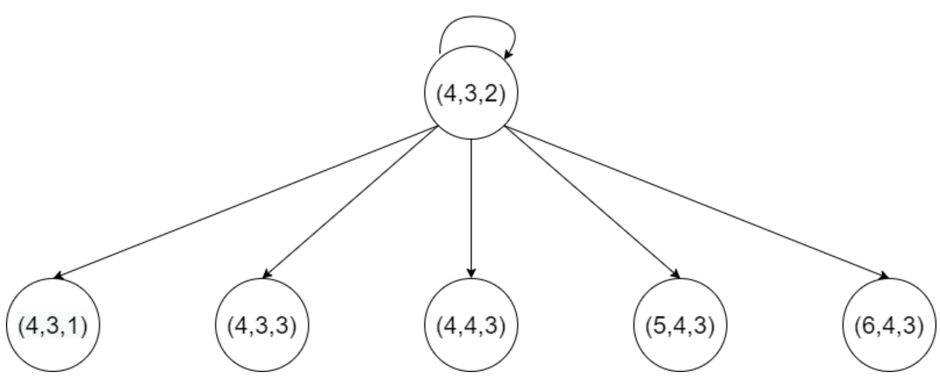

Figure 1: A section of the complete epistemic model. This section shows only the number of worlds that can be accessed when the current dice combination is $(4, 3, 2)$. All other worlds are excluded from this figure. However the complete model is an expension of this current subsection for every other combination. Every accessiblity is identical for all agents.

Recall that when a die is not rolled, it is shown to the other players. In the epistemic approach, this regarded as a public announcement and therefore these dice become common knowledge (Ditmarsch, van der Hoek, & Kooi, 2007, pp. 72-74). This can be reflected in the formula as a public announcement of (4,3,z) which lets all other agents know that the four and the three are present in the next real world. Equation 1 shows how the knowledge changes for all the agents.

$$[(4,3,z)]K_m(4,3,1) \vee K_m(4,3,2) \vee K_m(4,3,3) \vee K_m(4,4,3) \vee K_m(5,4,3) \vee K_m(6,4,3) \tag{1}$$

This formalization of the game allows for the knowledge agent to build a knowledge base of possible worlds and to reason based on the accessible worlds throughout the game.

2.3 Knowledge-based agent

One of the agents in the simulation will play the phases in its turn according to a more sophisticated strategy, involving knowledge and probabilities in the game. In contrast to the naive strategy agents, this knowledge-based agent will make decisions on the basis of available knowledge. In the game, any unrolled dice during the rolling phase should be put on top of the cup for all players to see. In the implementation, this means the unrolled dice are made available as common knowledge. Consequently, the knowledge-based agent can use this common knowledge to cross-out some of the possible worlds. The remaining possible world will be part of its knowledge and can be used for reasoning about its own decisions.

In the following part, the knowledge-based strategy in each of the three phases will be discussed. For each phase, we will explain how the agent uses knowledge for making a decision in that phase. This will be based on a regular turn (i.e. not a first turn of the game).

2.3.1 Phase 1: Believing or calling a bluff

First of all, the agent has to make the decision whether it believes the previous players bid or calls it a bluff. Now a description will be given of how the knowledge agent determines to call a bluff.

The agent will first evaluate whether the previous bid is possible with respect to its knowledge, by cross-checking whether the bid is within its set of possible worlds. If this is not the case, the agent will regard the previous bid as a bluff. Otherwise, the assessment will be done on the basis of probabilities. In that case, the agent will make a *joint probability distribution* (referred to JPD after this) of the possible worlds, where each possible worlds gets the probability for it to be rolled assigned to it. This probability depends both on the available knowledge, as well as on the number of combinations that make up for the roll, since there is only one possibility to roll (1,1,1), but multiple possibilities to roll (6,5,1). Hence, the probability P(Possible World | Knowledge $\wedge$ Number of Combinations) is determined as follows:

- **Common knowledge: 2 open dice:** Only one dice is unknown, meaning that there exist six possible worlds (in conjunction with the common knowledge). Every world that is impossible given the common knowledge can be disregarded and will not be used in the JPD, since their probability is equal to 0. Since there is only one possible way for each value to be rolled, each of the possible worlds will have a $\frac{1}{6}$ probability assigned to them in the JPD.
- **Common knowledge: 1 open dice:** There are two unknown dice. Again, every possible world in which the open die is not present can be disregarded and will not be used in the

JPD, since these are impossible. Furthermore, there are two possibilities for the unknown dice, which each have a different probability. These are the following:

(1) If both dice have an equal value, there is only one combination possible for worlds in which this is the case. This means these probabilities will be equal to $\frac{1}{6}^2 = \frac{1}{36}$.

(2) If the unknown dice have an unequal value, there are two possible combinations of dice rolls that make up for the same possible world (E.G. when 5 is common knowledge, both (5,1,4) and (5,4,1) will be (5,4,1)). Consequently, the probabilities assigned to these worlds in the JPD will be equal to $2 * \frac{1}{6}^2 = \frac{2}{36}$.

- **Common knowledge: no open dice:** In this case all three dice are unknown, meaning that the possibilities are determined by three unknown factors. Now all the possible worlds have a probability assigned to them in the JPD. Here we have the following three possibilities:

  (1) Each of the three unknown dice have an equal value. In these cases the assigned probability in the JPD equals $\frac{1}{6}^3 = \frac{1}{216}$.

  (2) Two of the three unknown dice have the same value, but one does not. In these cases there are 3 possible combinations of dice rolls that make for the same possible world (E.G. (2,1,1), (1,2,1) and (1,1,2) will all be (2,1,1)). Hence the assigned probability in the JPD equals $3 * \frac{1}{6}^3 = \frac{3}{216}$.

  (3) All dice have a different value. In these cases there are 6 possible combinations of dice rolls that make up for the same possible world (E.G. (6,5,4), (6,4,5), (5,6,4), (5,4,6), (4,6,5) and (5,4,6) will all be (6,5,4)). Consequently the assigned probability in the JPD equals $6 * \frac{1}{6}^3 = \frac{6}{216}$.

Now that the JPD is constructed, the agent can determine the probability that the previous players bid is actually rolled. This is equal to the cumulative probability of the worlds in the JPD that are equal or higher to this bid (since higher dice rolls than the bid are also not a bluff).

Following, a believe threshold has to be set, which determines whether the agent will call a bluff or not. It was found that this threshold should not be static, since realistically it happens that the same bid in the same situation is sometimes believed and sometimes not. Also the common knowledge is a factor in determining roll probability, hence that should also be included in determining the believe threshold. Otherwise the knowledge agent will either believe too many bids or call too many bluffs. Therefore the believe threshold is determined as follows:

- **Common knowledge: 2 open dice:** The believe threshold is equal to a randomly picked value from a normal distribution with $\mu = 1/4$ and $\sigma = 1/12$.

- **Common knowledge: 1 open dice:** The believe threshold is equal to 0.75 times a randomly picked value from a normal distribution with $\mu = 1/4$ and $\sigma = 1/12$.

- **Common knowledge: no open dice:** The believe threshold is equal to 0.5 times a randomly picked value from a normal distribution with $\mu = 1/4$ and $\sigma = 1/12$.

Given that these thresholds are determined with some randomness, the knowledge agent will have a probability to both believe and not believe a variety of bids. This seemed like the most elegant solution, since when a static threshold was used, the agent's decision would be deterministic.[3]

[3] Note that this believe threshold will always involve some kind of arbitrariness, which is due to uncertainty. Hence, finding the correct normal distribution is quite hard. However, this is also the case for human player.

Finally, when the probability retrieved from the JPD, given the bid and possible worlds, is lower than the believe threshold, the agent will call a bluff. In all other cases, if the knowledge agent believes the previous player not to be bluffing, it believes the bid and goes into the rolling dice phase.

2.3.2 Phase 2: Rolling dice

The knowledge agent has to decide which dice to roll (and which not to roll) in order to get a higher value than the previous bid. The following procedure is executed:

1. The agent will roll the lowest die first, since always at least one die has to be rolled. If this leads to a higher roll value than the previous bid, the knowledge agent will stop rolling and go to the bidding phase. Otherwise step 2 is executed.

2. If the unrolled dice are equal, there is a small chance of $\frac{1}{6}$ that the agent decides to go to the bidding phase and bluff on a poker, with the value of those open dice. This is because the agent reasons such that other players will see both open dice of that value as common knowledge when the agent stops rolling, hence making the bluff believable. If the dice are unequal, bluffing on the poker is not a viable option and step 3 is executed.

3. If there is at least one 6 in the current bid, move to step 4. Otherwise, if there is no 6 in the current bid (and it is not a poker), then the agent will try to roll one of the remaining dice to get a 6, since that has a reasonable chance to get the agent to a truthful higher bid. If still no 6 is rolled after this, the agent will roll the final die and try to get a 6. Also, if the agent had rolled a 6 already with the first try (and with that reached a higher possible bid), there is still a $\frac{1}{2}$ chance to roll the other die anyway, in order to try and reach a higher bid with it.

4. In this case there is exactly one 6 in the current bid (for two 6's go to step 5), and the agent has not reached a higher roll value than this bid with its first roll. Now the agent will determine whether it wants to roll another die, on the basis of what it has to roll in order to reach a higher bid, in accordance with the probability of rolling that die higher. This probability is equal to $\frac{1}{6}$ times the number of possibilities that are higher for that die. To clarify: when the die has value 1, there are 5 possibilities to roll the die higher than that 1, hence the probability becomes $\frac{5}{6}$. However, when the die is a 5, there is only one possibility to roll the die higher than that 6, causing the probability to become $\frac{1}{6}$. If the probability causes the knowledge agent not to roll the die, it will argue that bluffing with the common knowledge for the next agent is less risky than trying to roll higher.

5. Finally, in this case there are two 6s in the current bid, and the agent has not reached a higher roll value than this bid with its first roll. The agent will go to the bidding phase and bluff on poker 6.

After executing (a number of) these steps, the agent will go into the bidding phase.

2.3.3 Phase 3: Bidding

In this step the knowledge agent decides how to bid smartly. Smartly in this sense means that the agent will use the common knowledge available for the next agent to determine what is a believable bluff. This is done with the following procedure:

1. Firstly, the knowledge agent will always bid truthfully, if it has a higher valued cup than the previous bid after the rolling phase. Otherwise, go to step 2.

2. In this step, a bluff is made on a poker, given that this was decided during the rolling phase. This possibility was the case in steps 2 and 5 of the rolling phase. If this was not decided, a bluff must be made in the following step.

3. A number of possible bluffs are determined on the basis of the common knowledge for the next player. This is done by excluding bids that are impossible given the common knowledge. Given the number of possible bluffs, one that is a small random number higher than the current bid is chosen. For example, a (6,6,2) will have the knowledge agent choose randomly between (6,6,3), (6,6,4) and (6,6,5), given that the common knowledge allows this.

The performance of and agent executing this knowledge-based strategy will be compared to agents using the naive strategies.

3 Experiments

The performance of the knowledge agent was tested in a number of configurations, in which the strategies of the naive agents varied. In these experiments, the knowledge agent we focused on always was player 3. The game was ran 10000 times and the number of losses per agent was measured. Run times were approximately 30 to 120 seconds, depending on the configuration (the number of calculations for different strategies).

4 Results

In most of the configurations, the knowledge agent performed well and did not lose for a majority of the games. Figure 2 shows two of the results for configurations in which this is the case.

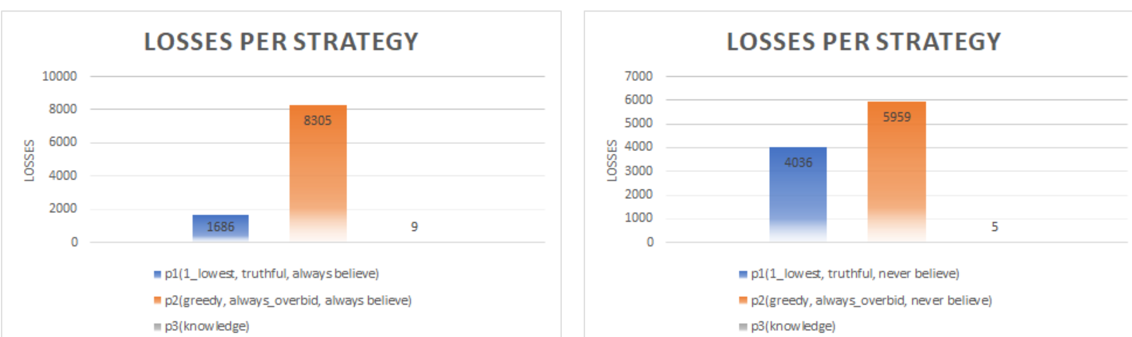

Figure 2: This Figure shows two of the configurations in which the knowledge agent performs well and loses only a few games. The strategies of the agents are shown behind the player number and depict (roll strategy, bid strategy, call strategy) respectively.

Some interesting results in which the knowledge agent performed worse, come from configurations in which it's preceding player is an agent applying the greedy rolling strategy, in conjunction with a truthful bidding strategy. Some results of these strategies can be found in Figure 3.

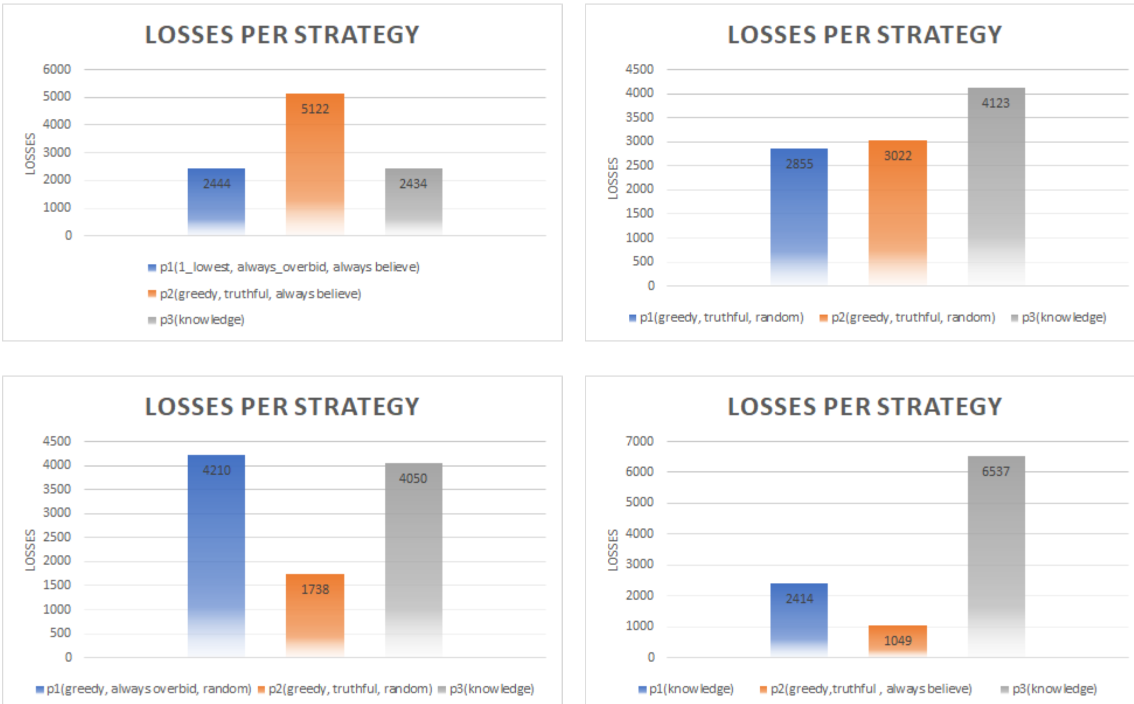

Figure 3: This Figure shows a number of the configurations in which the knowledge agent performed worse and loses a relatively large number. The strategies of the agents are shown behind the player number and depict (roll strategy, bid strategy, call strategy) respectively.

5 Discussion

From the results we can conclude that the knowledge agent performs well against most of the strategies. However, depending on the strategy before it, the greedy + truthful strategy has an advantageous position to roll in such a way that the knowledge agent has trouble assessing the bid. This can be explained by the fact that on the one hand a greedy + truthful strategy tends to perform quite well as a naive strategy in general. Essentially, this strategy tends to perform actions that maximize the possible values during rolling, whilst also being truthful about its bid when possible.

Furthermore, the knowledge agent can exactly determine the probability of the previous player's bid given the common knowledge. What the knowledge agent should do with this probability, is another issue. In order for the knowledge agent to make a decision, a believe threshold must be set to which the probability is weighed. We found that setting this threshold led to a trade-off. If the threshold is set too static or in a narrow range, the knowledge agent starts playing more or less deterministically. On the other hand, if the threshold is too wide, the agent might accept or reject certain bids that a human player would not accept. Balancing this threshold such that sometimes a bid is accepted and sometimes rejected can be quite hard. Given that the greedy + truthful strategy tries to roll 6's and announce them truthfully, the knowledge agent can come into the situation quite often in which the greedy + truthful strategy has rolled quite high, yet the probability extracted from the JPD is quite low. In the current implementation, the believe threshold is drawn from a normal distribution and only varies by means of the amount of common knowledge. In reality, a belief threshold varies per turn, given the risk of an action, but also information like mimics, body language and context. The absence of this information leads to limitations for the knowledge agent, especially when determining the belief threshold.

In order to make the knowledge agent utilize the knowledge to its full potential, future research can focus on improving the balancing of the believe threshold. Other work could focus on implementing more sophisticated behavior, such as including risk into the game, in which a player becomes risk-aversive near losing a game, or risk-seeking otherwise. Finally, the game can be extended with other game rules that were now left out due to simplifications. For example, the game now only implements rolling under the cup, such that unrolled dice become common knowledge. Also, the variant could be implemented in which dice can be rolled outside of the cup, leaving a number of dice from the previous player hidden. Knowledge-wise this would add an interesting aspect.

In this project, a model was made in which artificial agents play the game of blufpoker. A number of naive strategies were implemented, as well as a strategy based on knowledge and reasoning in the game, in which the knowledge is a result of epistemic logic. The knowledge agent performed well against most strategies, yet a strategy that rolled greedy and bid truthfully when possible competed against the knowledge agent. A GUI was made to visualise the game and the knowledge and reasoning of the knowledge agent.

References

Ditmarsch, H. v., van der Hoek, W., & Kooi, B. (2007). *Dynamic Epistemic Logic* (1st ed.). Springer Publishing Company, Incorporated.

Meyer, J.-J. C., & Hoek, W. V. D. (1995). *Epistemic Logic for AI and Computer Science*. Cambridge University Press.

6 Appendix: Graphical User Interface

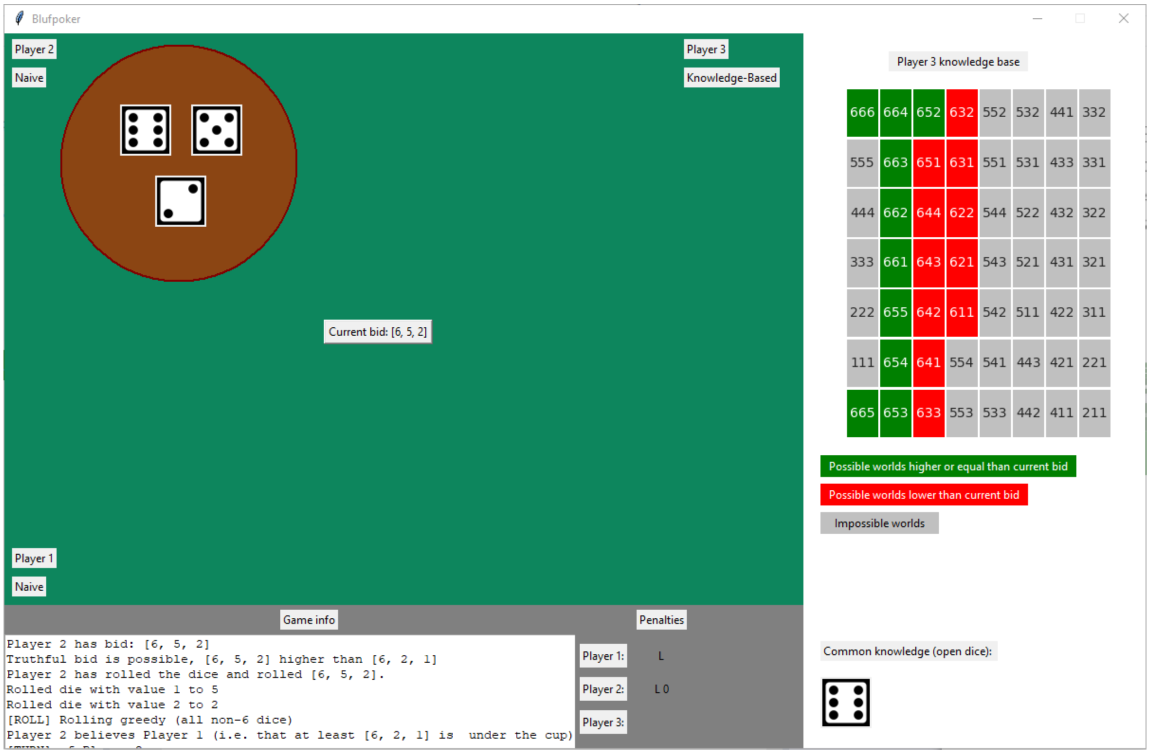

Figure 4: This Figure shows the graphical user interface. The central panel provides a game interface with depicted dice and players. The panel on the bottom shows additional game information in the form of a text box and the penalties belonging to the players. The panel on the right shows the common knowledge, and a graphical depiction of the deductions made by the knowledge agent in terms of possible worlds and how they relate to the current bid.