LAMAS Project:
Analyzing Blufpoker using Epistemic Logic

Ivar Mak (S2506580) Oscar de Vries (S2713748) Joost Warmerdam (S2591367)

1 Introduction

Blufpoker is a dice game which contains several interesting aspects with regards to knowledge, beliefs and bluffing. In this project, we aim to implement (a simplified version of) the game in a programming environment, as well as simulating different players and their respective strategies. This will be done using Epistemic Logic combined with a Multi Agent System that models playing behaviour for three players. We aim to implement different playing styles, which will be further explained in Section 2. Furthermore, we aim to define and compose an epistemic model of the knowledge for at least one player. This model will contain the possible worlds which can be reached from the in game situation and will be used for the decision making process of the player. The aspect of bluffing (false public announcements) comes into play when a player throws a bad turn. But before we address this, we will describe the rules of the game.

1.1 Game Description

The game is played using three dice, which 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 translates to the single digit. So in a situation where a 5, 3, and 2 are rolled the corresponding number is 532. The exception are the "pokers" which are triplets (i.e. 111, 222, 333, etc). These are the highest possible throws, 111 is higher than 665, and 666 is poker 6 and the highest possible score. Table 1 describes all possible combinations, ranked descending from highest to lowest. Players take turns throwing the 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 throw a number that is higher that the number that was passed on by the previous player. Each dice can be thrown once in order to try to reach a higher number. When the player completes his turn, he needs to publicly announce the number that he reached which needs to be higher than the number that was passed on to him (unless he is starting the round, in this case any number suffices). 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 saying a number that he did not throw. Since the dice are concealed, he can say whatever number he deems suitable, and pass on the dice to the next player. It is up to this player to determine whether his adversary is telling the truth. If he believes the truth was told, it is his turn to throw the dice and 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. Note that not all dice have to be thrown in order for a player to pass the dice, a player can stop and announce his number after each throw. Furthermore the number of rounds to be played is variable, with a player obtaining one penalty point for each round lost, and the game ending when a specific number of points is reached. 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", receiving a letter for each loss.

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: Set of possible dice combinations in the game of *Blufpoker*, ranked from top left to bottom right, descending per column.

1.2 Simplifications

The original game consist of multiple aspects that will heavily complicate the composition of the model and therefore the decision making process. This is why we will apply several simplifications to make the game more manageable.

- When a player is not bluffing, he will announce the exact number he reached with the dice.
  In the original game, the player can choose whichever number he wants (even 'impossible' numbers such as 500) as long as his throw is higher than this.
- The "golden triangle" is not the highest throw.
  In the original game, the combination 321 yields a "golden triangle" which is the highest possible throw, beating 666.
- All players play to win.
  In the original game, players can decide to adjust their announcements in order complicate specific other player's turns, trying to make them lose. In our version the player is only concerned with himself not losing.
- The player will never pass the dice without throwing.
  In the original game, a player has a third option besides throwing the dice or calling a bluff. Which is announcing a higher number, while not having thrown the dice, and passing them to the next player. This does not add to the knowledge base of the player, which is why we will not implement this part.

1.3 Outline

We aim to build an algorithm that simulates the game, in which agents play according to different strategies. These strategies will include superficial strategies, but also a more complex strategy based on the knowledge that is obtained. Given this approach, we aim to address the following question: How does an agent with a knowledge based strategy perform versus more naive, superficial strategies?

In the following section we will describe formally how we aim to construct our knowledge base, and how we base our decision making process on this. Furthermore, we will describe how the naive players achieve their decision making. In the Results section, we will describe which results we will gather and how we will utilize these in order to draw conclusions. Depending on the progress of the project, we will try to implement extensions with regards to the baseline version of our game, as well as the model and decision making. The directions for these extensions are described in the Section Possible Extensions.

2 Methods

The Blufpoker simulation made in Python to model the game and the epistemic knowledge model will be used to create a strategy for the knowledge-based agent.

2.1 Simulation

A simulation of Blufpoker is made to quantify the performance of the knowledge-based agent. In this simulation, the rules described in section 1.1 are implemented in a setup for a three-player game of Blufpoker. This means there are three agents that play against each other, each with their own strategy. One of these agents will use the knowledge-base for its decision making (Section 2.2). The simulation will perform the game in a step-wise manner and assign penalty letters accordingly. Once, an agent has reached the maximum number of penalty points, that player has lost. The simulation will record, which player has lost as well as data about the playstyle of the knowledge-based agent. Relevant data about the knowledge-based agent are the number of bluffs in one game, the ratio of bluff-calls per game.

The strategies of the other 'naive' agents will differ in three manners: (1) deciding which dice they roll, (2) deciding whether they will bluff, and (3) deciding whether the previous player bluffed. These strategies are not implemented yet but will be used to create results about how well the knowledge-based agent performs against these different strategies.

2.1.1 Naive Strategies

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

1. Which dice will the agent throw?
    - Random
    - Throw the lowest die
    - Throw any die that is lower than 6
2. When will the agent bluff?
    - Always be truthful, if possible.
    - Always tell a number x above the previous value
3. When has the previous agent bluffed?
    - Random
    - Always believe it
    - Always call the bluff

2.2 Knowledge-based agent

(Oscar) One of the agents in the simulation will play according to a more sophisticated strategy, involving knowledge and probabilities in the game. We will now go over the 3 phases in the game that involve decision making and explain how the knowledge-based agent will apply its strategy. This will be based on a regular turn (i.e. not a first turn of the game).

2.2.1 Phase 1: Believing or calling a bluff

First of all, the agent has to make the decision whether it believes the previous players bet or calls it a bluff. This will be done by looking at the number of possible worlds in which the previous player is bluffing. If the percentage of these worlds (in which the bet is not true) is bigger than a threshold, the model will call a bluff and the cup will be lifted.

If the previous player is not believed to be bluffing, the agent believes the bet and goes into the throwing dice phase.

2.2.2 Phase 2: Throwing dice

The model has to decide which dice to throw (and which not to throw) in order to get a higher value than the previous bet. It does so by calculating the percentage of possible worlds for which this is the case. From this it can be calculated which dice are best to throw first and whether its best to throw another die. Note that in contrast to the possible worlds in phase 1, the possible worlds here represent the *expected* possible worlds given a certain action, namely that of throwing a particular die.

After the player is happy with the dice its rolled, or can not roll any more dice, we go into the bidding phase.

2.2.3 Phase 3: Bidding

For the bidding phase there are two options. Either the agent can bid truthfully to what its value is, or it could bluff and bid a value higher than its cup. Explaining the truthful bid is self-explanatory, it's simply stating the value the agent has under its cup.

Bluffing is one of the hardest steps to implement, since it usually involves higher order knowledge. For bluffing the agent could make use of (a combination of) the following strategies:

Always bet slightly higher than the previous bet. This could be a random number higher than the value of the previous bet, but different every time it is applied.

Determine which bluffs are most believable, according to previous players turns. This could involve some memory, for example when player 1 throws (6,5,1), then player 2 throws one dice and calls (6,5,3) and player 3 throws one dice and calls (6,5,4). At this point, both player 1 and player 2 can know with large certainty that player 3 was speaking the truth. For player 1, it is the case that he knows that player 2 has a high belief of what is approximately on the table. If player 1 decides to throw just one dice (E.G the '4'), then he knows that player 2 holds the states (6,5,5) (6,6,4) and (6,6,5) for possible as higher values than the previous bet. However, since players rationally tend to throw their lowest dice first, the option (6,6,4) could be less viable than the other options. Hence, if player 1 has not thrown enough for the bet to be on the table, he could throw again to make the bet, or bluff on either (6,5,5) or (6,6,5).

3 Results

The results we will gather are focused whether we will be able to efficiently construct a model and its accuracy. We will evaluate the decision making process based on this model by measuring the performance of our logic based agent compared to the other agents. Firstly we will measure this by looking at the ratio between won and lost games. Secondly, by looking at the amount of bluffs that are successfully called by our logical agent, drawing conclusions about in what way the knowledge in this model aids to making the correct decision.

4 Possible Extensions

If time allows for it, we have thought of several directions to take in order to extend our project and/or it's implementation.

- We want to evaluate different settings with regards to the strategies against which the logical player has to compete, and look at the way its performance changes.
- When possible, we would like to add more players using a logical model for their decision making process.
- We would like to add feedback prompts describing the decision making process of the logical agent, much like the thought process of a human player.
- More aesthetically, we would like to add visualizations of the dice and or the cup.