

---

# SOFCOMPUTING: FUZZY LOGIC USED IN SIMPLE CHECKERS GAME

---

A PROJECT REPORT

**Bartosz Gardziejewski**

Wroclaw University Of Science and Technology  
Computer Science, Internet Engineering  
226128@student.pwr.edu.pl

**Rafał Grądkowski**

Wroclaw University Of Science and Technology  
Computer Science, Internet Engineering  
194479@student.pwr.edu.pl

January 20, 2020

## ABSTRACT

Ranging from artificial intelligence to the control of engineering systems, a number of different applications relies on fuzzy logic as an important tool. In this project, a simple fuzzy logic model was proposed to study its performance as an artificial player in a simulated checkers game implemented using Python language capabilities.

**Keywords** Softcomputing · Fuzzy logic · Checkers · Python · scikit-fuzzy

## 1 Introduction

Fuzzy logic had been studied since 1920s, notably by Łukasiewicz and Tarski [1], as *infinite-valued logic*, because the term itself was only proposed by Lotfi Zadeh in the 1965 with his *fuzzy set theory* [2, 3].

It is an approach to computing that, rather than using the usual true or false boolean logic, is based on an idea of 'degrees of truth', where the truth values may be any real number between 0 and 1 both inclusive [4]. This form of many-valued logic is employed to handle the concept of partial truth, most often deduced from imprecise and non-numerical information. Fuzzy models have the capability of recognising, interpreting, utilising, manipulating and representing data and information that are vague and lack certainty.

Checkers (also known as Draughts) is a strategy board game for two players which involve diagonal moves of uniform game pieces and mandatory captures by jumping over opponent pieces. Nowadays, many versions of the game exist, however only one specific was chosen to be implemented for purposes of the project - American checkers, played on an 8x8 checkerboard.

## 2 Checkers

American checkers, played on 8x8 checkerboard [Fig. 1], is played by two opponents on opposite sides of the gameboard. Only the dark squares of the checkerboard are used. One player has the dark pieces, the other the light pieces. Players begin with twelve pieces each and the player using light pieces starts the game. Players alternate turns and cannot move an opponent's pieces. A move consist of moving a piece diagonally to an adjacent unoccupied field. If the adjacent field contains an opponent's piece, and the field immediately beyond is vacant, the piece must be captured (removed from the game) by jumping over it. The player, who cannot move because of being blocked or without pieces remaining, loses the game. When a piece reaches the farthest row forward it is marked as a 'Queen piece' and it acquires additional powers, including the ability to move and capture backwards [Fig. 2].

Artificial player in project's implementation always starts as second, playing with dark pieces.

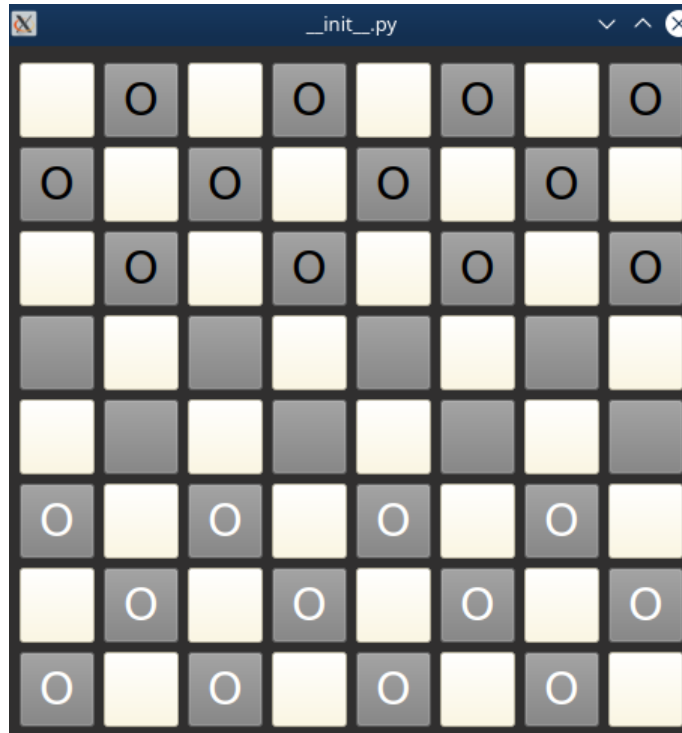


Figure 1: Basic view of the checkers game implemented for the purpose of this project, showing placement of the pieces at the beginning of a new game.

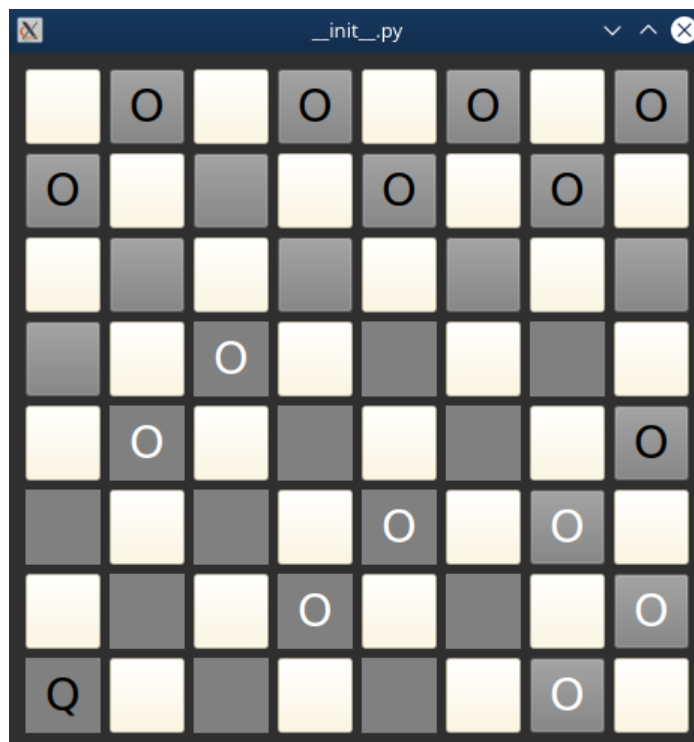


Figure 2: Checkerboard with dark Queen piece visible in bottom left corner.

### 3 Fuzzy logic model

Fuzzy logic model proposed in this paper consists of separate fuzzy control systems [FCS] linked by simple IF statements. Each FCS tries to resolve a problem that resembles a commonly known *Tipping Problem* [Fig. 3]. IF statements control the flow of data in the model to optimally choose next move in the game based on results returned by each FCS.

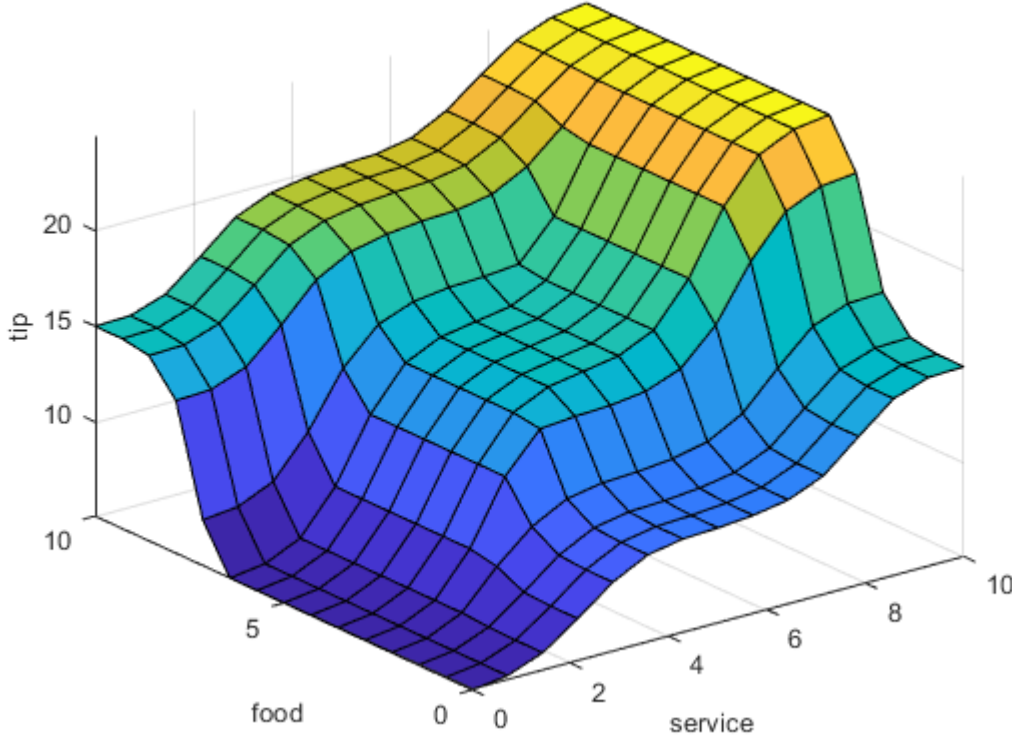


Figure 3: Control space of an exemplary Tipping Problem.

#### 3.1 Terminology

A fuzzy variable has a *crisp value* which takes on some number over a pre-defined domain called a *universe* in fuzzy logic terms. The crisp value is how the variable can be described using normal mathematics. For example, taking previously mentioned Tipping Problem as a base for explanation in this section, if fuzzy variable was to describe how much to tip a waiter in restaurant, it's universe could be 0 to 25% and it might take on a crisp value of 15%.

The fuzzy set, which can be used to describe the “fuzzy value” of a fuzzy variable, is a set of terms used to describe the variable. These terms are usually adjectives like “low,” “slow,” and “good.” Each term has a membership function that defines how a crisp value maps to the term on a scale of 0 to 1. In essence, it describes “what” something is.

Commonly, three types of the membership functions are used: triangular, trapezoidal and gaussian. In the project, triangular function was used for all membership functions [Fig. 4].

Therefore, a “good tip” might have a membership function which has non-zero values between 15 and 25%, with 25% being a “very good tip” (e.g., it's membership is 1.0) and 15% being a “almost a good tip” (e.g., its membership is 0.1).

A FCS links fuzzy variables using a *set of rules*, which are simply mappings that describe how one or more fuzzy variables relates to another. These are expressed in terms of an IF-THEN statement; the IF part is called the *antecedent* and the THEN part is the *consequent*. In the tipping example, one rule might be “IF the service was good THEN the tip will be good.”

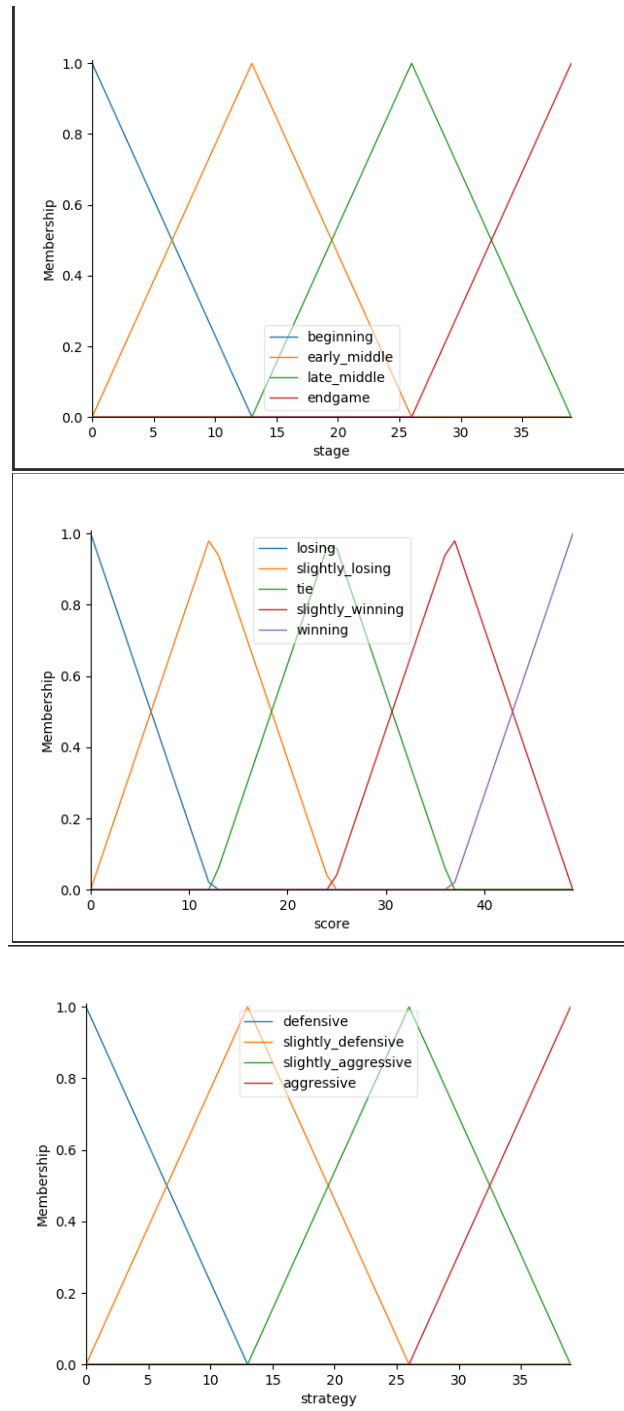


Figure 4: Membership functions used to determine player's strategy.

### 3.2 Strategies

Four strategies were designed to choose from: defensive, slightly defensive, slightly aggressive and aggressive. Each strategy has its own rules determining how player's possible moves are prioritized. Based on these rules, each move from a list of possible moves is given some weight, used to sort the list and choose the best one from strategy's point of view. Strategy is checking only one turn ahead, which means that weight evaluation knows that moving piece cause losing it or capturing enemy piece in next turn and nothing beyond that. The move weights in strategies are evaluated as follows:

- **Defensive:**  
Concentrates on pushing pieces that are behind, avoids capturing enemy pieces if it mean losing a piece, keeps Queens out of the game if possible.
- **Slightly defensive:**  
Concentrates on keeping a balance between moving pieces in front and in the back of the board, avoids capturing enemy pieces if it mean losing a piece.
- **Slightly aggressive:**  
Concentrates on keeping a balance between moving pieces in front and in the back of the board, captures enemy pieces even if it mean losing a piece, pushes Queens towards the center of the board.
- **Aggressive:**  
Concentrates on pushing pieces in front towards enemy side, captures enemy pieces even if it mean losing a piece (including Queens, if it will cause multiple captures of enemy pieces), pushes Queens toward center of the board.

The weights in those strategies may be tweaked for betterer perform, although values chosen seem to work properly when connected with fuzzy logic strategy determiner.

### 3.3 Model's flowchart

Each FCS used in the model [Fig. 5] and all variables describing them were presented in details in below subsections.

#### 3.3.1 Determining game's stage

In this particular case, if number of passed turns exceeds value of 200, it's artificially set to be equal to 200. Rules described on Fig. 6 were applied to FCS to determine game's stage, which resulted in control space that can be seen on Fig. 7.

- Antecedents
  - Number of passed turns
    - \* Crisp value range: 0 - 200
    - \* Fuzzy set: [Low, Medium, High]
  - Number of pieces held by losing player
    - \* Crisp value range: 0 - 12
    - \* Fuzzy set: [Low, Medium, High]
- Consequents
  - Game's stage
    - \* Crisp value range: 0 - 40
    - \* Fuzzy set: [Beginning, Early middle, Late middle, Endgame]

#### 3.3.2 Determining artificial player's score

Rules described on Fig. 8 were applied to FCS to determine player's score, which resulted in control space that can be seen on Fig. 9.

- Antecedents
  - Number of player's pieces

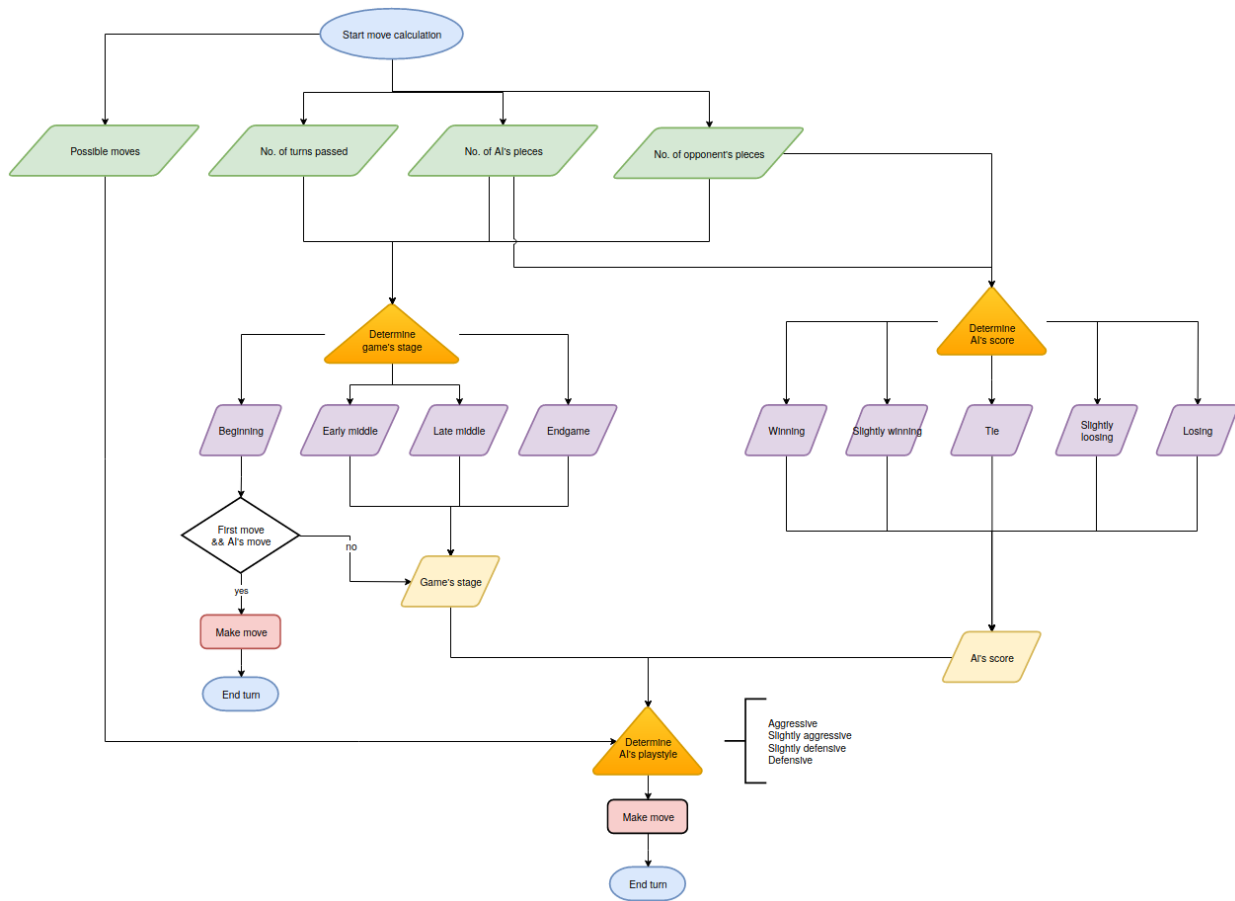


Figure 5: Artificial intelligence player decision flowchart with fuzzy logic elements.

		Stage		
Number of turns	Low	Beginning	Early middle	Late middle
	Middle	Early middle	Late middle	Endgame
	High	Early middle	Late middle	Endgame
		High	Middle	Low
		Number of pieces		

Figure 6: Rules applied to FCS to determine game's stage.

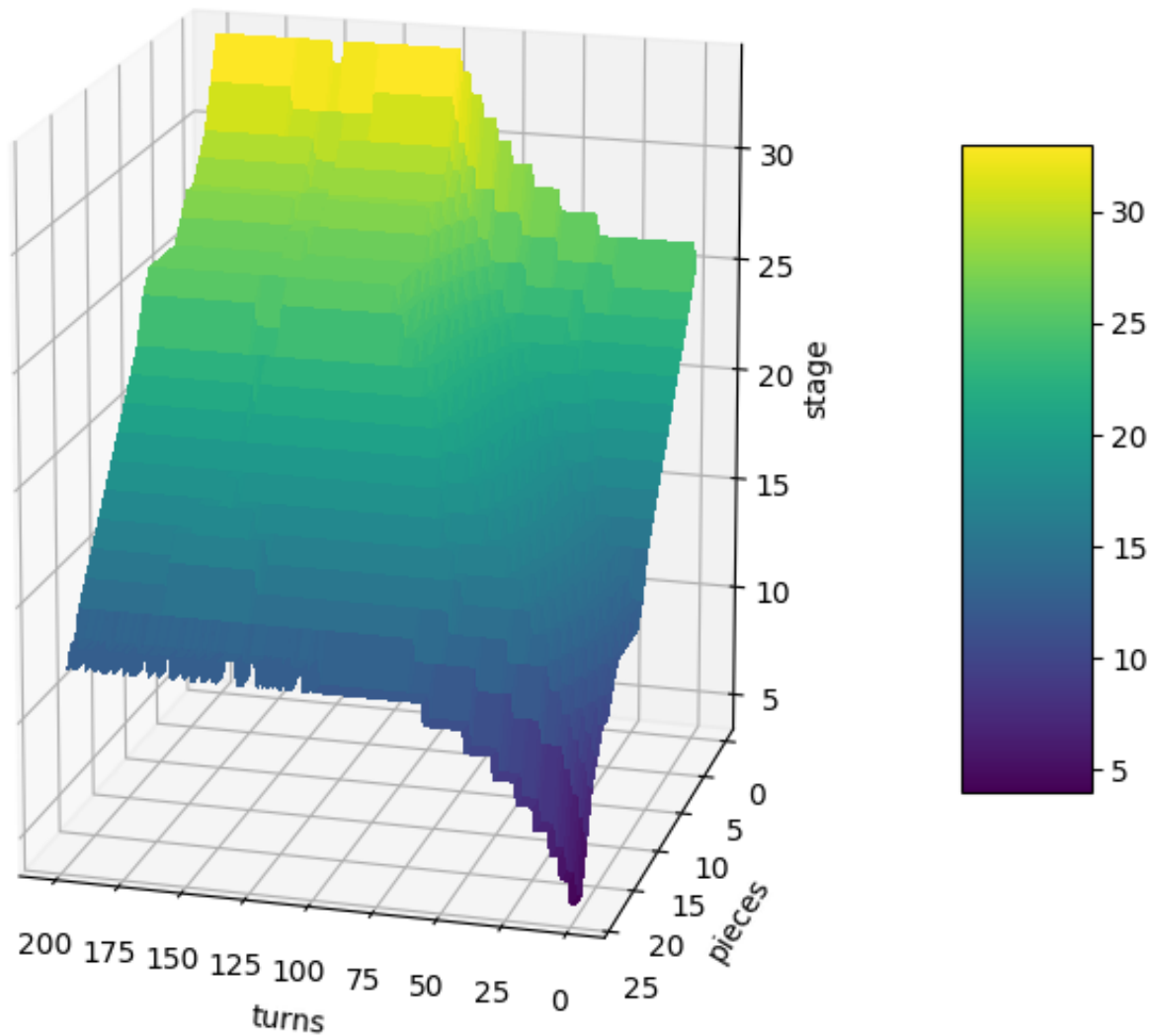


Figure 7: Control space 3D plot of the system responsible for determining stage of the game.

- \* Crisp value range: 0 - 12
- \* Fuzzy set: [Low, Low medium, High medium, High]
- Number of opponent's pieces
  - \* Crisp value range: 0 - 12
  - \* Fuzzy set: [Low, Low medium, High medium, High]
- Consequents
  - Player's score
    - \* Crisp value range: 0 - 50
    - \* Fuzzy set: [Losing, Slightly losing, Tie, Slightly winning, Winning]

### 3.3.3 Determining artificial player's strategy

Rules described on Fig. 10 were applied to FCS to determine player's strategy, which resulted in control space that can be seen on Fig. 11.

- Antecedents

		Score			
Number of pieces	High	Losing	Losing	Slightly losing	Tie
	Medium high	Losing	Slightly losing	Tie	Slightly wining
	Medium low	Slightly losing	Tie	Slightly wining	Winning
	Low	Tie	Slightly wining	Winning	Winning
		Low	Medium low	Medium high	High
Number of pieces					

Figure 8: Rules applied to FCS to determine player's score.

- Game's stage
  - \* Crisp value range: 0 - 40
  - \* Fuzzy set: [Beginning, Early middle, Late middle, Endgame]
- Player's score
  - \* Crisp value range: 0 - 50
  - \* Fuzzy set: [Losing, Slightly losing, Tie, Slightly winning, Winning]
- Consequents
  - Player's strategy
    - \* Crisp value range: 0 - 40
    - \* Fuzzy set: [Defensive, Slightly defensive, Slightly aggressive, Aggressive]

### 3.4 Results

Proposed sets of inputs and rules, although natural from average human perspective, especially for determining player's score and stage of the game, didn't perform as well as expected. Artificial player lacked any dynamism, preferring slightly defensive and slightly aggressive strategies, while pure defensive and aggressive strategies were rarely chosen [Fig. 12]. Tweaking multiple model's values was giving promising results, however simultaneously decreasing performance in other areas, e.g. choosing purely defensive / aggressive strategies rather than their 'slight' variations.

## 4 Conclusions

Despite checkers are considered a solved game since 2007 [5], which in game theory means that a game's outcome can be correctly predicted from any position, assuming that both players play perfectly, the model proposed in this project certainly does not achieve a perfect play. However, it can be definitely stated that the model can be a good opponent for a casual player.

This implies that choosing a fuzzy logic controller system for gaming purposes can be a good choice, as it is easy to design and implement, but only when achieving casual player's level of expertise is considered fitting. To tune



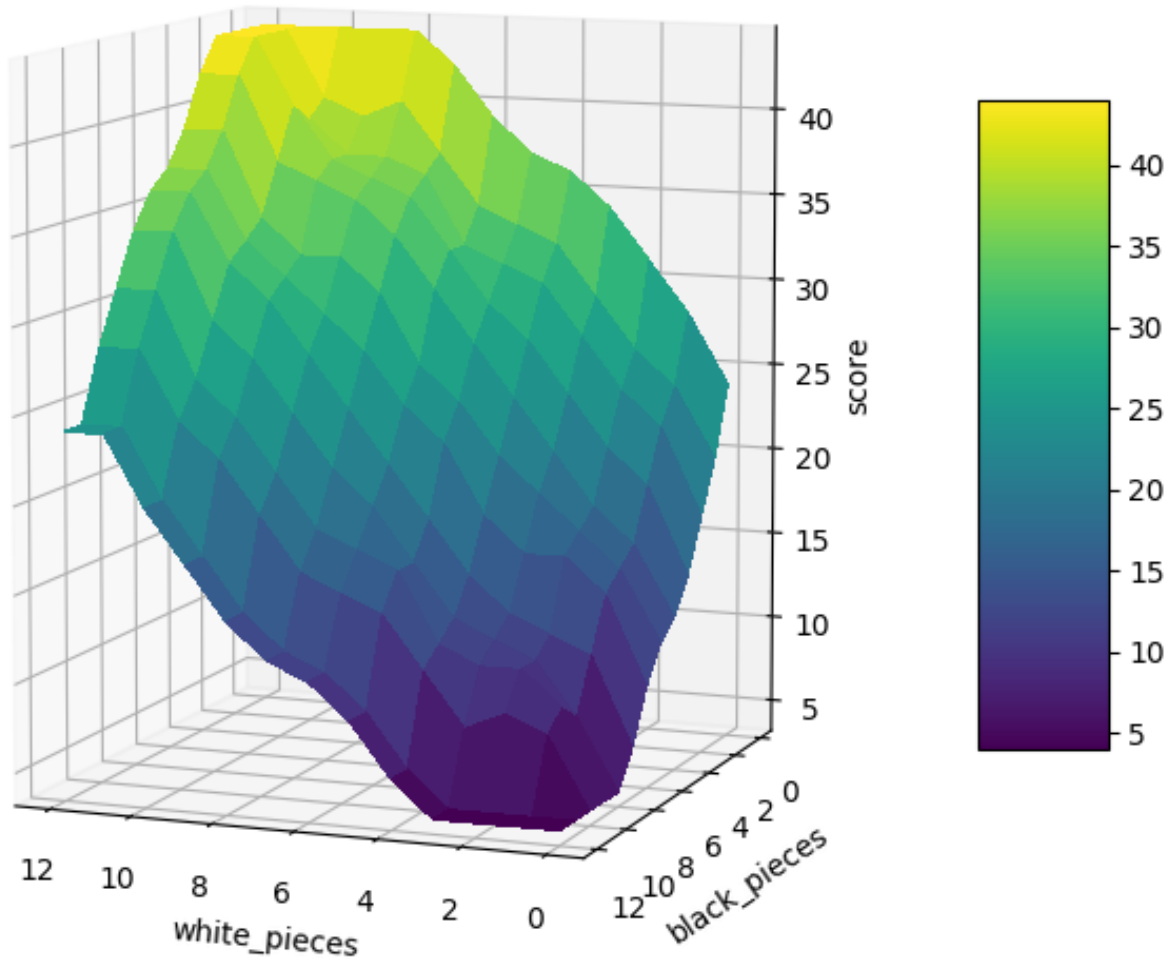


Figure 9: Control space 3D plot of the system responsible for determining player's score.

such model to make better decisions could be considered too time consuming because all of model's flowchart, inputs, outputs, rules and membership functions must be engineered solely by hand.

## References

- [1] Francis Jeffry Pelletier. Petr hájek. *metamathematics of fuzzy logic*. trends in logic, vol. 4. kluwer academic publishers, dordrecht, boston, and london, 1998, viii 297 pp. *Bulletin of Symbolic Logic*, 6(3):342–346, 2000.
- [2] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338 – 353, 1965.
- [3] Fermüller Christian G. Cintula, Petr and Carles Noguera. Fuzzy logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, fall 2017 edition, 2017.
- [4] J. Mockor Vilém Novák, Irina Perfilieva. *Mathematical Principles of Fuzzy Logic*. Springer US, 1999.
- [5] Jonathan Schaeffer, Neil Burch, Yngvi Björnsson, Akihiro Kishimoto, Martin Müller, Robert Lake, Paul Lu, and Steve Sutphen. Checkers is solved. *Science*, 317(5844):1518–1522, 2007.

		Strategy				
Stage	Endgame	Defensive	Slightly defensive	Slightly aggressive	Aggressive	Aggressive
	Late middle	Defensive	Slightly defensive	Slightly defensive	Slightly aggressive	Aggressive
	Early middle	Defensive	Slightly defensive	Slightly defensive	Slightly defensive	Slightly aggressive
	Beginning	Defensive	Defensive	Slightly aggressive	Slightly aggressive	Slightly aggressive
		Losing	Slightly losing	Tie	Slightly wining	Winning
		Score				

Figure 10: Rules applied to FCS to determine player's strategy.

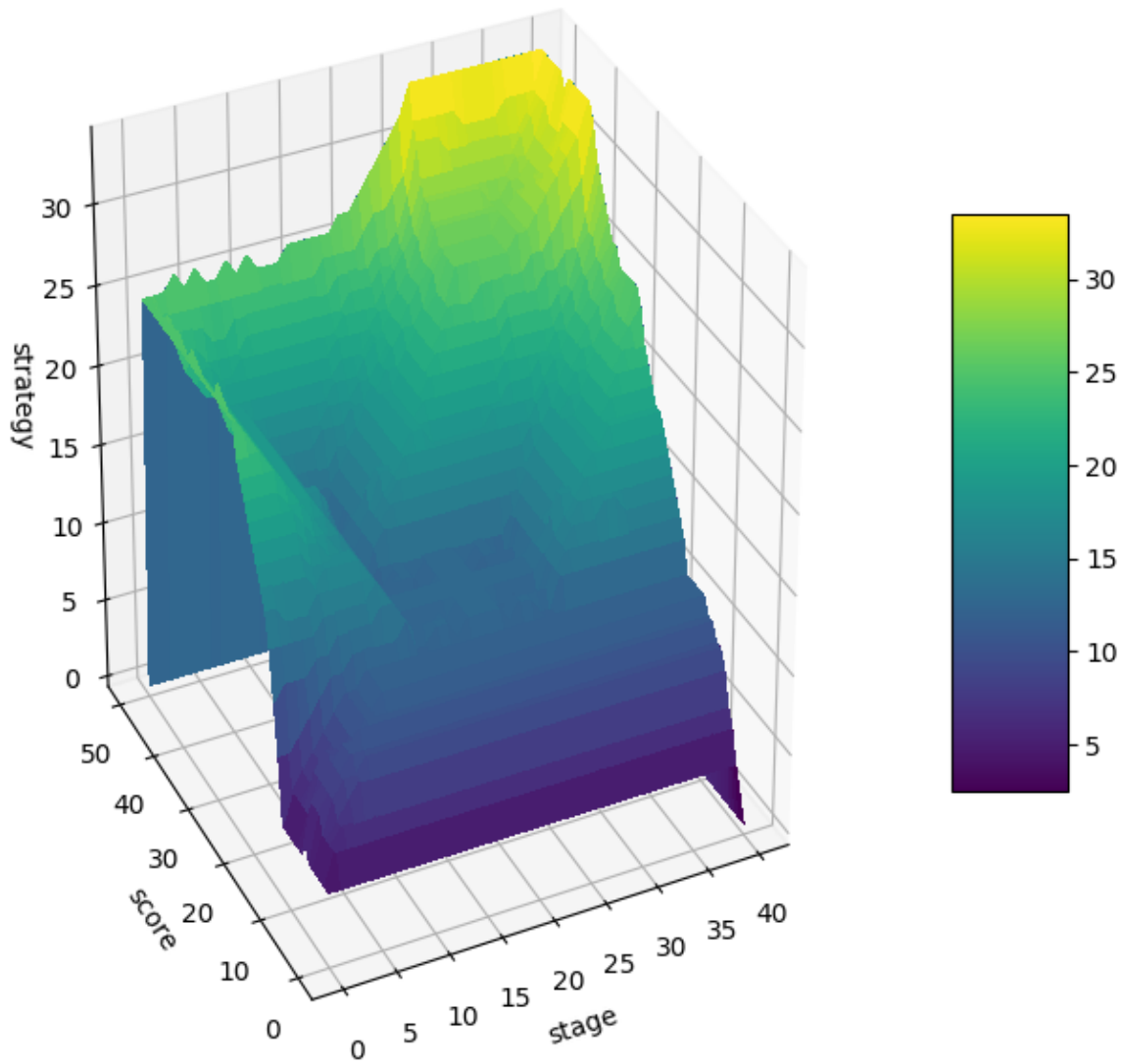


Figure 11: Control space 3D plot of the system responsible for determining player's strategy.

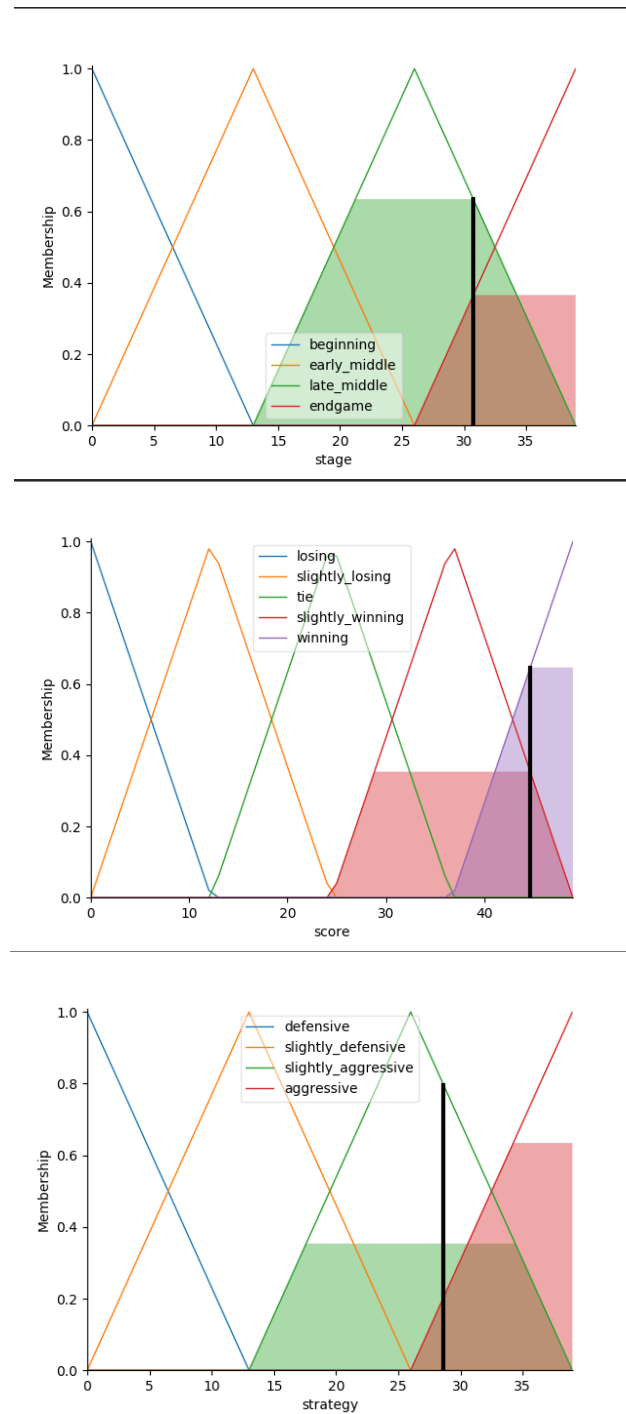


Figure 12: Determining 'slight aggressive' strategy when pure 'aggressive' strategy should be chosen according to rules specified on Fig. 10.