Fuzzy Goal-Driven Strategy for the Game "Geister"

Fall 2018 | ECE 4870/7870 –CS 4770/7770: Introduction to Computational Intelligence

Supervised by: Dr. James M. Keller

- Imad Eddine Toubal
- Olha Kholod
- Aidan Verhulst
- Shayne Wadle

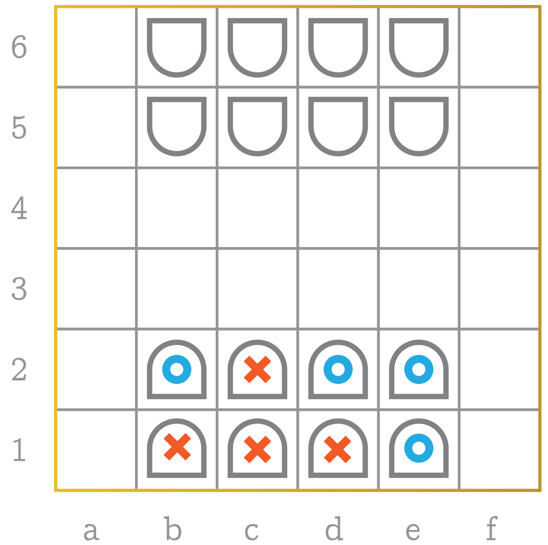
Mizzou
University of Missouri

INTRODUCTION

This project introduces an approach to design a winning strategy for the game of Geister, which utilizes fuzzy logic inference along with evolutionary optimization.

Keywords—fuzzy logic, evolutionary optimization

Geister Rules



Geister game board, only the player's own ghosts are known to be either good or evil.

Geister (German for Ghosts) is a strategic board game that takes place on a 6x6 square grid representing a haunted castle. Each player have to arrange four evil ghosts. The nature of each ghost is marked on the back of the figure and is hidden to the opponent player

Each turn a player can choose to move one of his/her ghosts one step away, such as forward, backward, left or right, but never diagonally. When landing onto an opponent's ghost, the latter is captured by the current player and true ghost's nature is revealed.

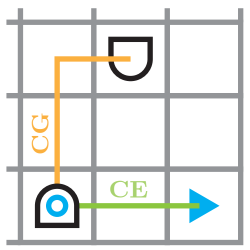
In order to win, a player has to reach one of the following goals:

- Capture all the good ghosts of the opponent player;
- Have all evil ghosts captured by the opponent player;
- Move one of her/his good ghosts off the board from the opponent's corner squares (marked with an arrow).

LINGUISTIC VARIABLES

In order to use fuzzy logic, we first need to define the different linguistic variables. This includes the observable features that will act as an input to the Fuzzy AI agent, as well as actions that are considered outputs. Below is an example of such linguistic variables:

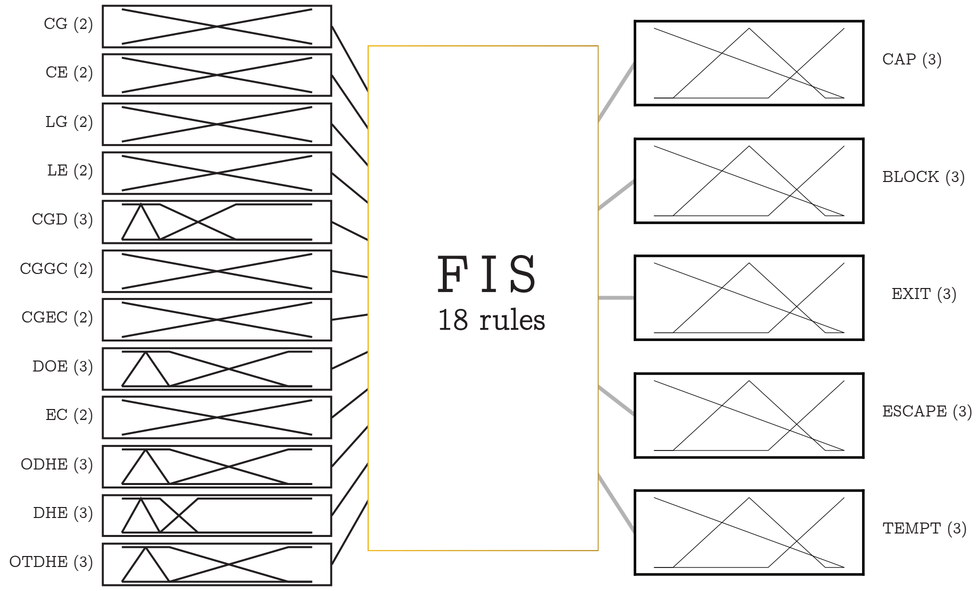
- **CG** [input]: the distance to the closest opponent ghoast.
- **CAP** [output]: the action to capture an opponent ghost.



Examples of used linguistic variables

Most features are specific to each ghost, but some are general (like the number of lost ghosts; LG)

FUZZY INFERENCE SYSTEM

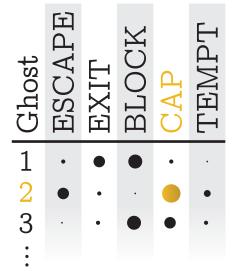


System geister

is: 12 inputs, 5 outputs, 18 rules

The FIS shown above was built with 12 linguistic variable inputs, and 5 outputs (actions). The output is determined by a set of 18 rules. Both input and output membership functions are illustrated in the diagram above.

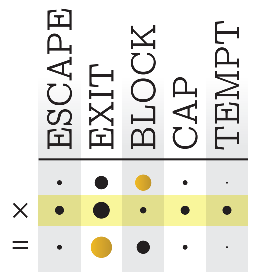
After generating an output for each of the AI Agent's ghosts, a utility matrix representing each ghost/action combination is generated. The naive way of chosing the ghost/action combination is to just find the maximum utility value in the matrix.



**Defuzzified output values are represented by the circle radius Maximum is highlighted in gold*

An interesting experiment, however, is to prioritize certain actions over others by giving weights to each action. How can we find the optimal values? Is this a problem we can solve using evolutionary approach?

EVOLUTIONARY COMPUTATION



The weights (chromosome) are highlighted in yellow

First, we assign weight values for the utility of each action. The array of weight values was encoded into the chromosome. The fitness function for each chromosome is the outcome after playing a number of games (winning likelihood) given the set of weights of that chromosome.

GA Hyper Parameters

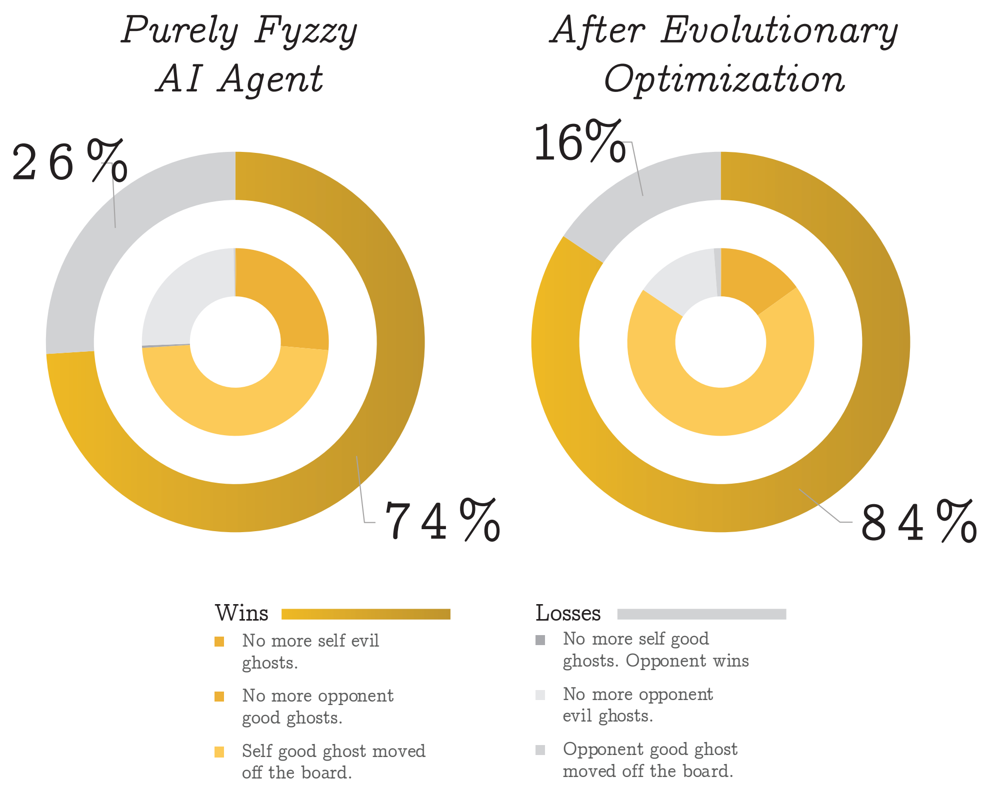
50	10%	5	30
Population size	Mutation Rate	Parents used in crossover	Number of Generations

RESULTS & ANALYSIS

After 30 genetation, below is our champion chromosom, we can notice that the numbers are not so far off from 1 still. However, a more significant change is observed in the performance.

ESCAPE	EXIT	BLOCK	CAP	TEMPT
1.07	1.03	1.45	1.05	0.90

This could be interpreted intuitively as the following: "block more often and tempt less often".



Results after playing 1000 games for each agent (Purely Fuzzy, and Evolutionary Optimized) against a random agent.

Analysis

Given the results visualized in the pie-charts above, two major conclusions can be drawn:

- The Purely Fuzzy system performs pretty well on its own (75% win rate)
- Prioritizing actions through weights plays a significant role in how well the AI agent performs.
- Our evolutionary algorithm has successfully found an optimum set of weights that increases the winning likelihood.

Buck, A. R., Banerjee, T., & Keller, J. M. (2014, July). Evolving a fuzzy goal-driven strategy for the game of Geister: An exercise in teaching computational intelligence. In Evolutionary Computation (CEC), 2014 IEEE Congress on (pp. 28-35). IEEE.