Elevens Strategy

Shreyas Muppana

1/11/2023

1 Introduction

Elevens is a single-player card game that is played using a standard deck of fifty two cards (excluding Jokers). The objective of the game is to clear a board of nine cards by forming pairs that add up to eleven.

The game begins with a set of cards placed face up on the board. The board should have nine cards at the start of the game. The player's task is to select cards from the board (to make a play) and form pairs that add up to eleven. For example, a player can select a five and a six, or a three and an eight. When a pair is formed, the cards are removed from the board. The player can also play a Jack, Queen, and King, in which case, all three cards will be removed. Cards are drawn from the shuffled deck to replace. The player wins when there are no more cards left in the deck, but if there are no valid plays before that, the player loses.

2 Notation

If C is a numbered card, then C' is its complement. For example, if $C = 5$, then $C' = 6$, and if $C = 1$, then $C' = 10$.

If F is a face card, then F_1 and F_2, in no order, are the other two face cards that make up the Jack-Queen-King combination. For example, if $F = \text{J}$, then $F_1 = \text{Q}$ and $F_2 = \text{K}$ (or vice versa).

3 Calculating the Value of a Card

We must take into consideration these following factors when calculating the value of a card:

- To win a game, the player must make exactly twenty pairs, and exactly four Jack-Queen-King combinations
- At the beginning of a round:
  - Each numbered card has exactly four other numbered cards it can pair with, in hand or deck
  - Each face card has exactly sixteen Jack-Queen-King combinations it is part of, in hand or deck
  - There are sixty four possible face card combinations
- The *immediate* value of a card will depend on:
  - For numbered cards:
    - The number of cards on hand that equal C'
  - For face cards:
    - The number of face cards on hand that equal F_1
    - The number of face cards on hand that equal F_2
- The *future* value of a card will depend on:
  - For numbered cards:

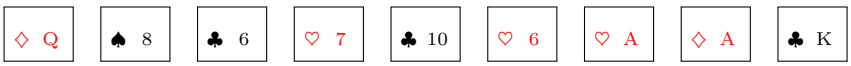
    - The number of cards in the deck that equal C'
  - For face cards:
    - The number of face cards in the deck that equal F_1
    - The number of face cards in the deck that equal F_2

Keeping the above in mind, we can calculate the value of a card as follows:

- Where $D(C)$ = number of cards in the deck that equal C
- Where $H(C)$ = number of cards on hand that equal C
- Where n = number of cards in the deck
- For numbered cards:
  - For immediate value, $I(C) = H(C')$
  - For future value, $F(C) = \frac{D(C')^2}{n}$
- For face cards:
  - For immediate value, $I(F) = \frac{H(F_1)+H(F_2)}{2}$
  - For future value, $F(F) = (\frac{D(F_1)+D(F_2)}{2})^2 * I(F)$

Finally, $V(C) = I(C) - F(C)$.

For example, a starting state ($n = 43$):



4 Strategy

The strategy is simple:

- Play the card with the highest value ($V(C)$)

  - If there is a tie, choose one randomly

Continuing the previous example, we can calculate the values of all of the cards. In this case, we would play (not just because it is the only move, but because it is the best) the ten along with either of the aces.

5 Simulation

In the elevens_simulator directory, there is a Rust program that simulates a game of elevens with random choices and with strategic choices. You will need to have Rust installed to run it. Adjust n in src/main.rs to change the number of samples to take. The number of games actually played will be anywhere from 14 to 22 times n, since a single sample is basically playing games repeatedly until a game is won. Some experimental results:

- $n = 1000000$
  - Random: On average, 9.358113 games needed to be played to win
  - Strategic: On average, 9.275664 games needed to be played to win

Large values of n take a while to run, especially with strategic choices, as the program is relatively unoptimized (one of the reasons I chose Rust, I wanted it to be somewhat fast without overt optimization). It's recommended to spin up a cloud instance or VM (i.e. EC2 or Github Codespaces) to run the program, or run it locally if you have a powerful enough computer (or have time to wait).