Agent for Bazar Blot

G. Hovhannisyan A. Abrahamyan
M. Khachatryan, M. Davtyan

December 3, 2024

Table of contents

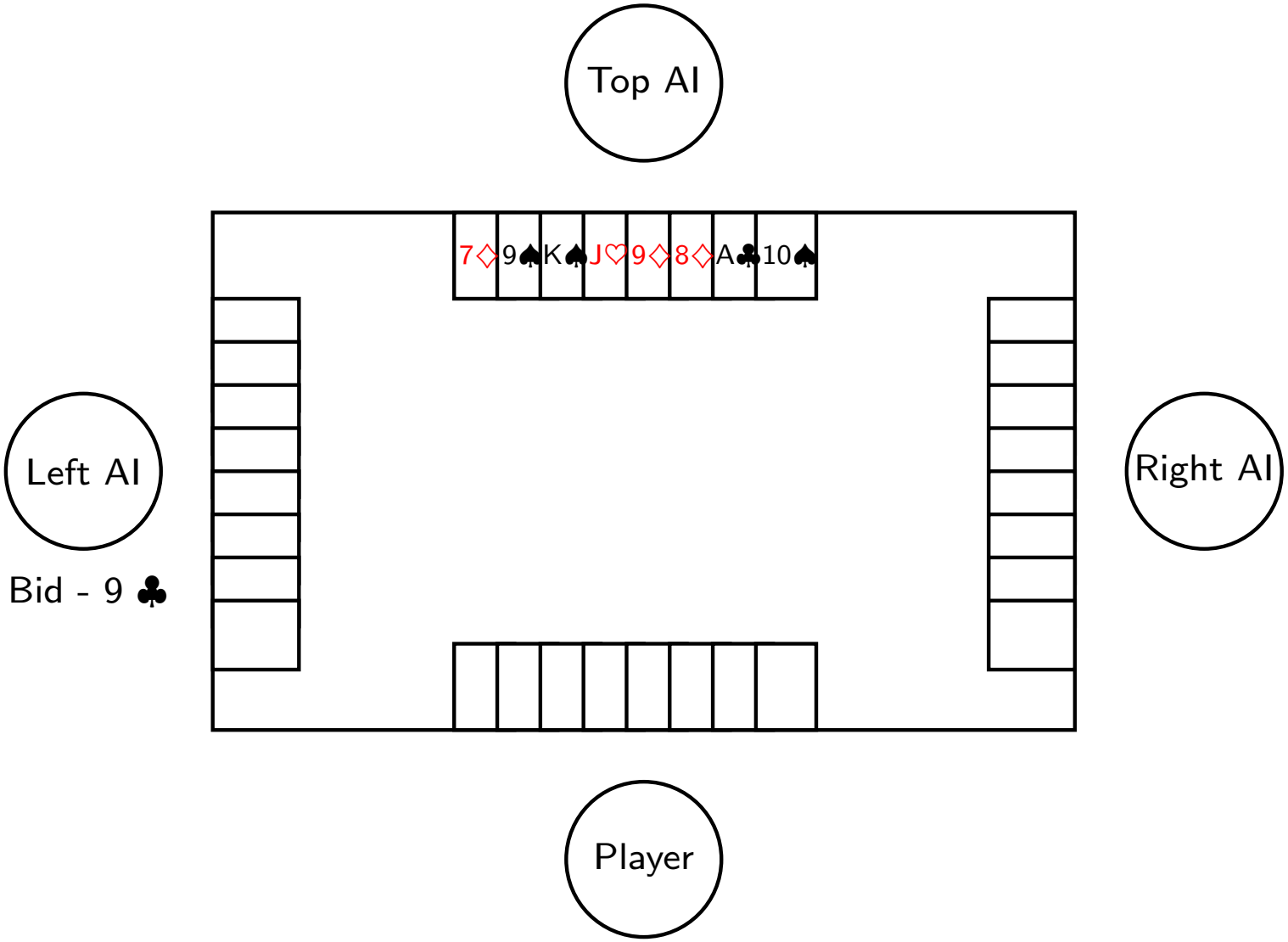
Introduction

Introduction

- The Origin of the Game
- General Rules
- Bidding
- Playing

Card	Trump	Regular	No-Trumps
A	11	11	19
K	4	4	4
Q	3	3	3
J	20	2	2
10	10	10	10
9	14	0	0
8	0	0	0
7	0	0	0

Introduction

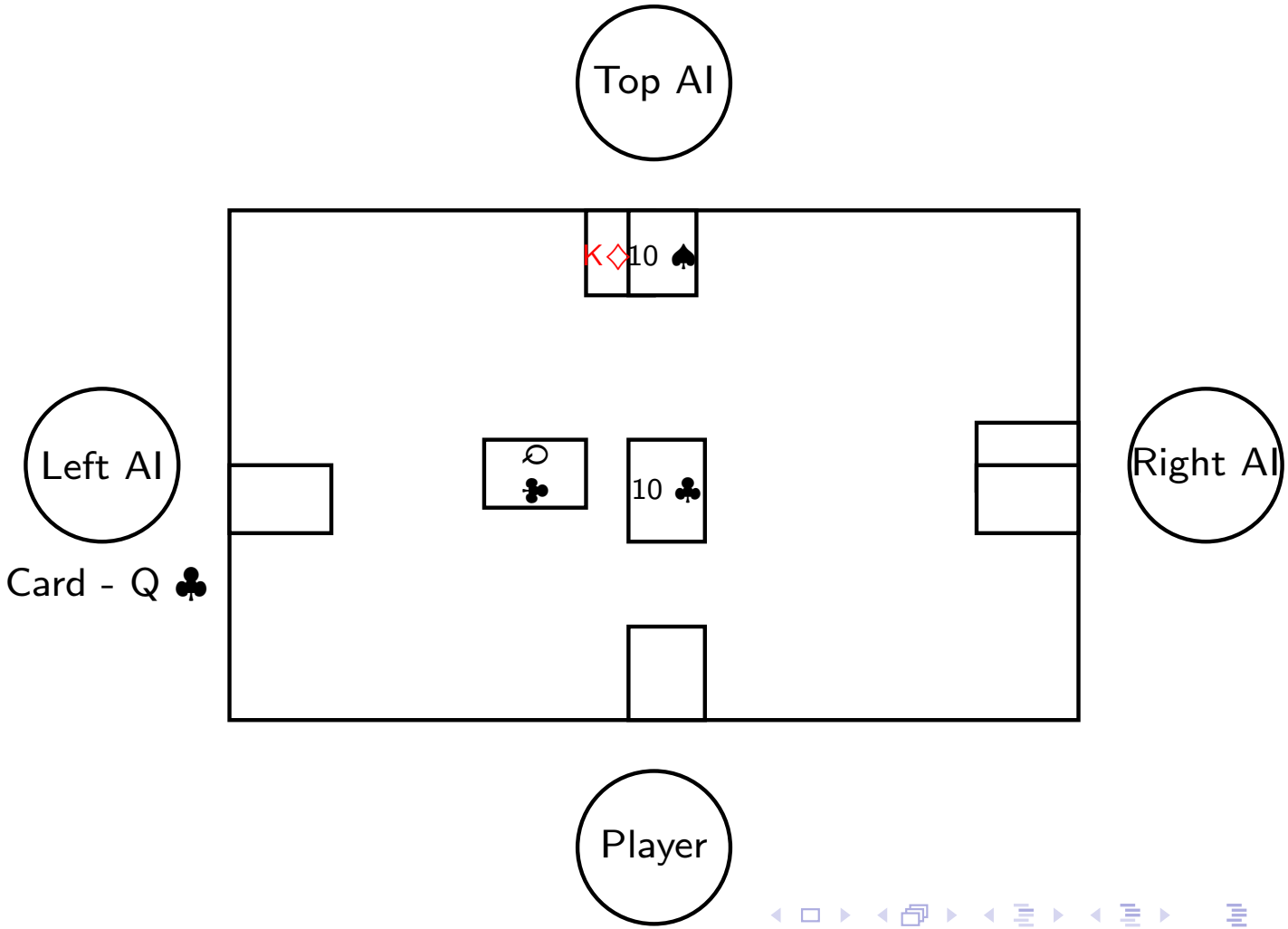


Cheating Mini-Max

Expectimax

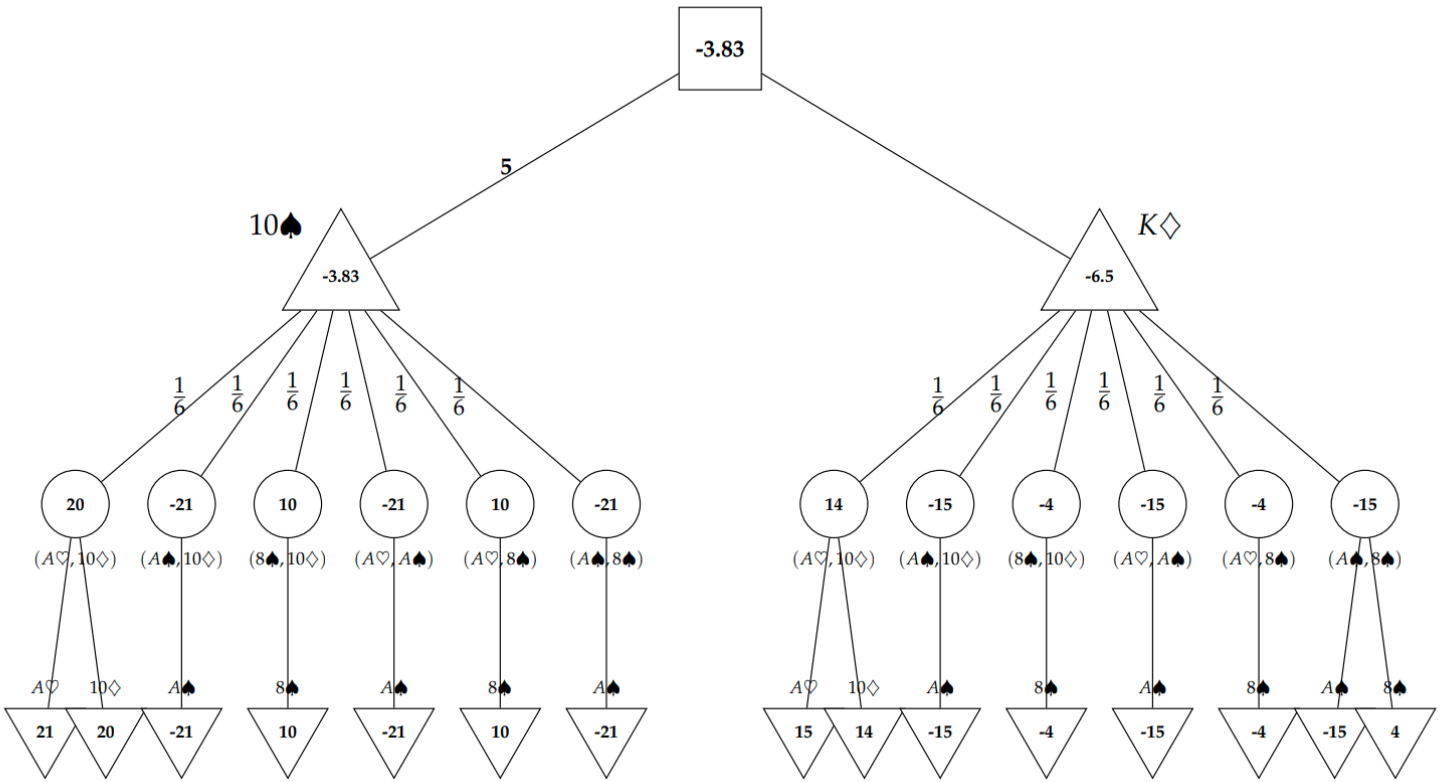
Without resorting to cheating we will have $\binom{24}{8}$ nodes in our first depth!

Figure: Example of a Game



Expectimax

Figure: Example of Expectimax tree



Mini-Max

- Assuming we know everyone's cards what's the approximate number of terminal states?

Mini-Max

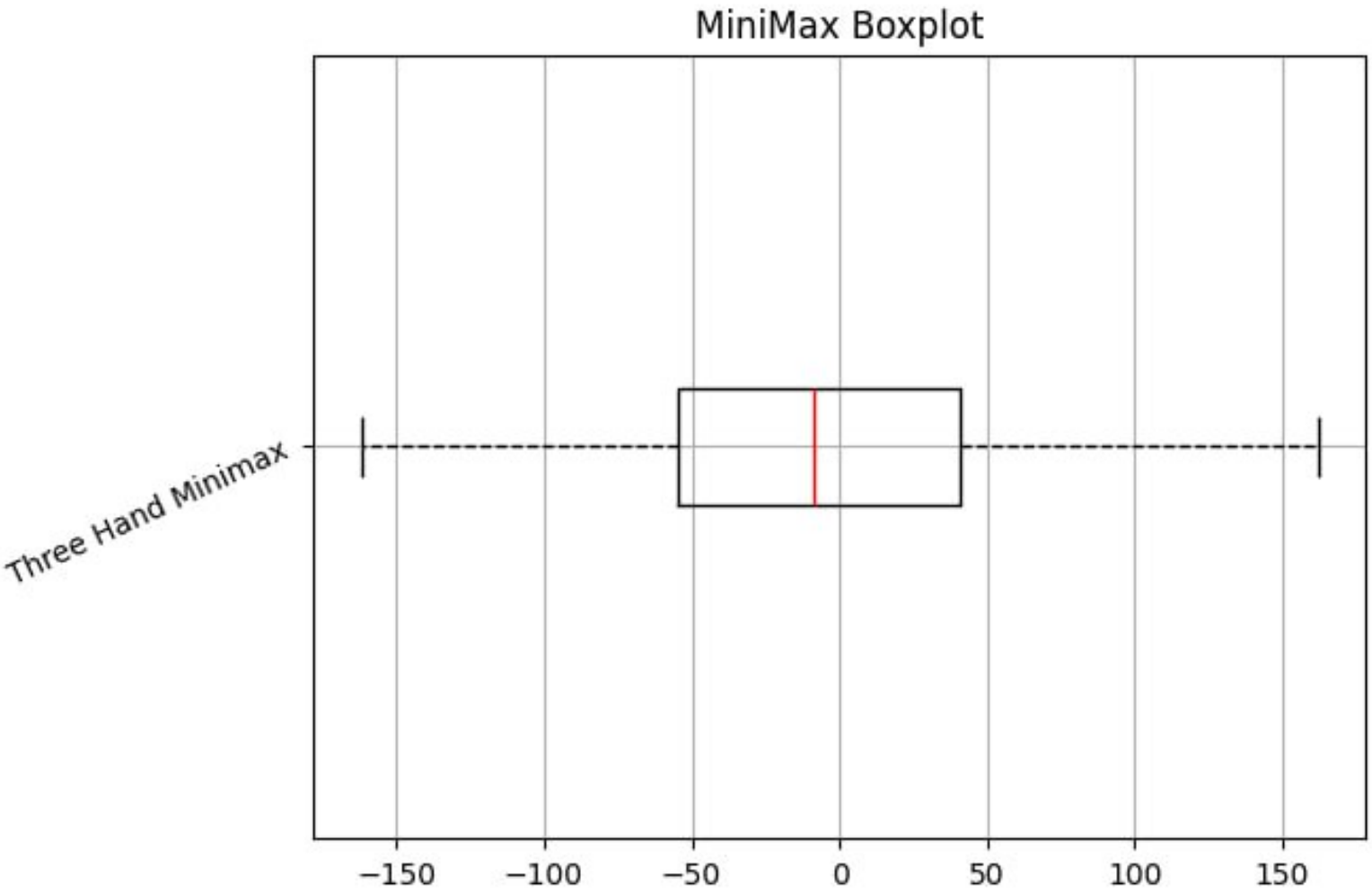
- Assuming we know everyone's cards what's the approximate number of terminal states?
- $(8!)^4$
- We can't use normal Mini-Max either even with alpha-beta pruning.

Depth-Limited Cheating Mini-Max

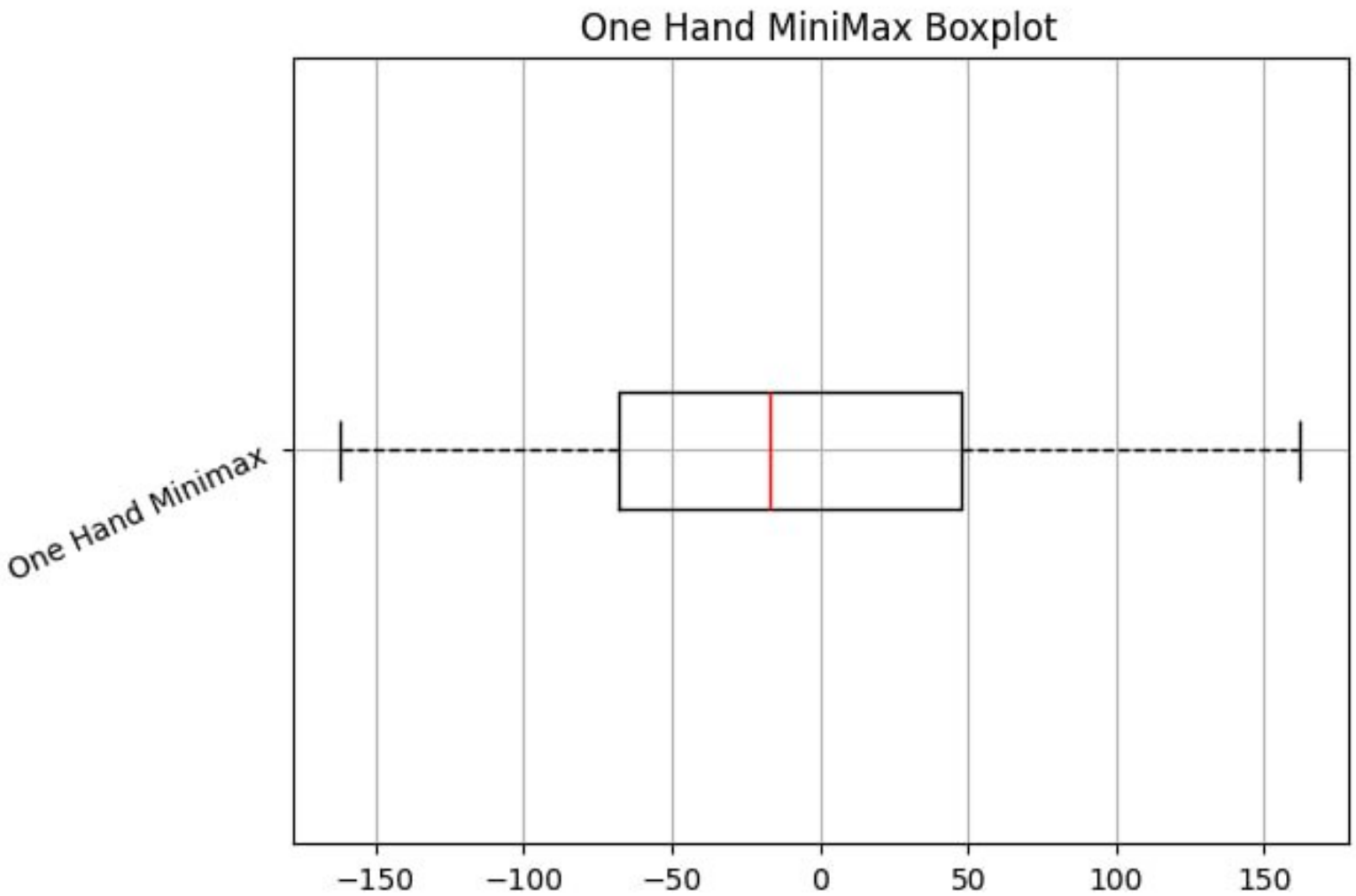
- Depth is limited to 12.
- Uses alpha-beta pruning.
- Heuristic is the current score / 10.
- It plays slightly worse than a random bot.

Simulation Results

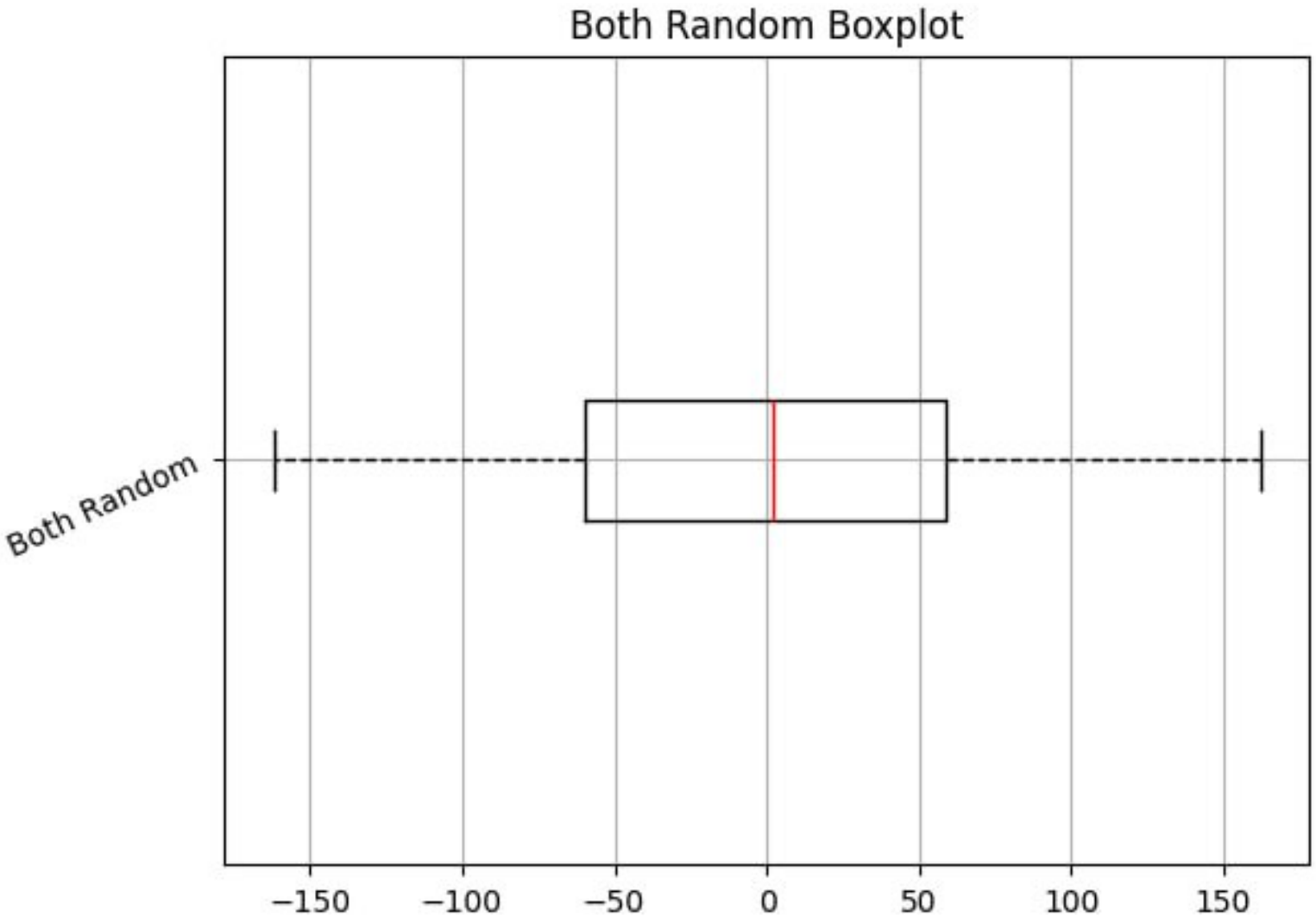
Minimax vs Random Without Auction (Three hand version)



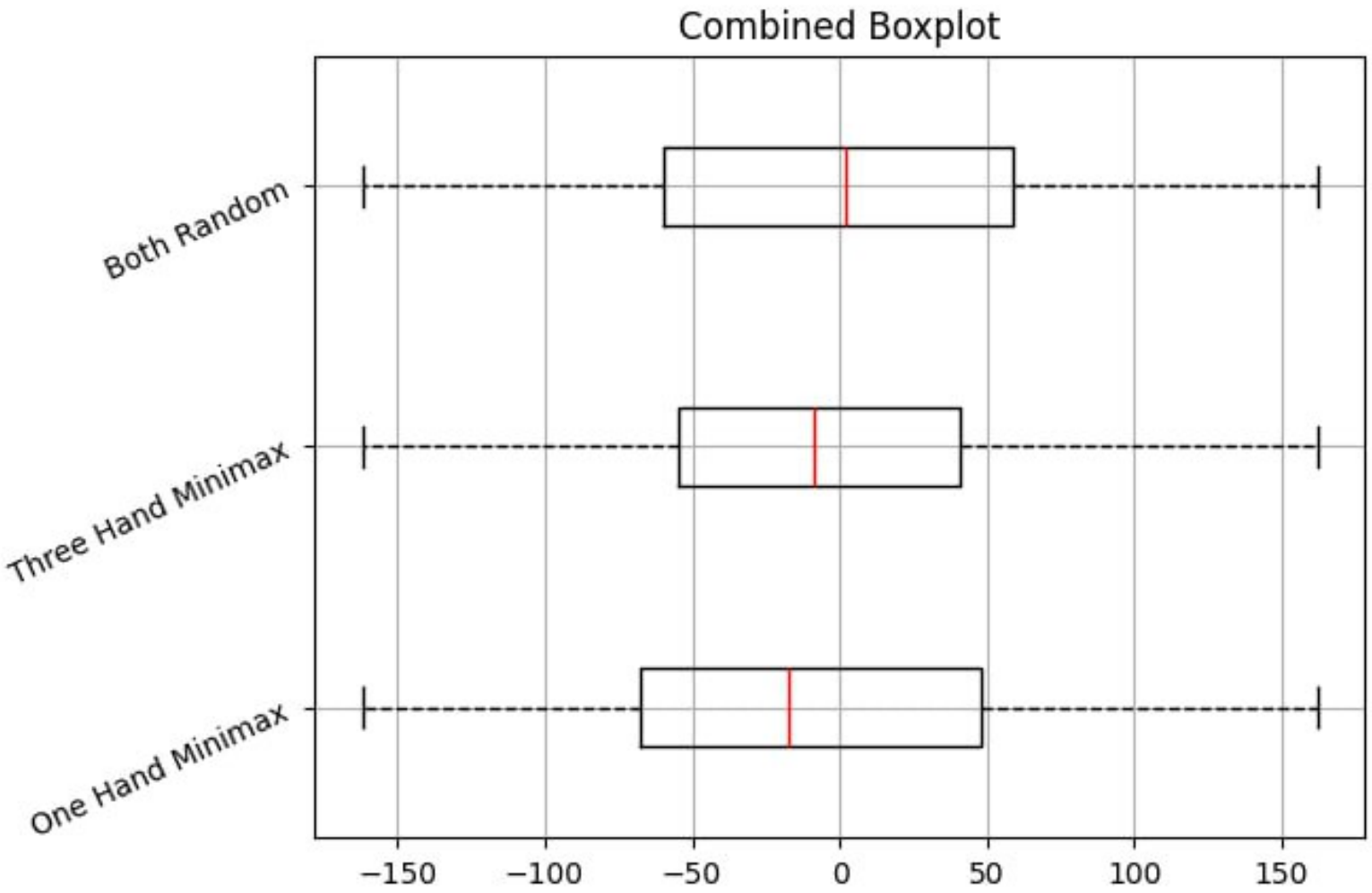
Minimax vs Random Without Auction (One hand version)



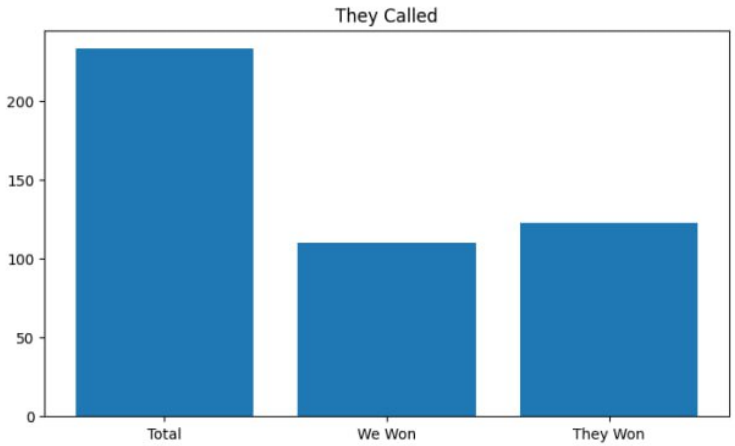
Random vs Random Without Auction (Control)

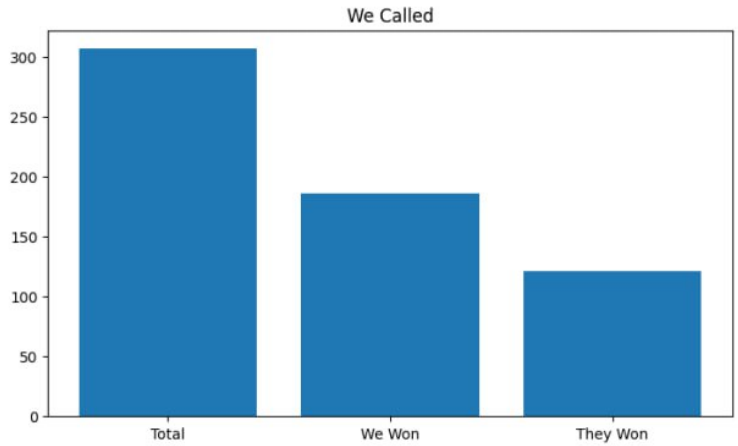


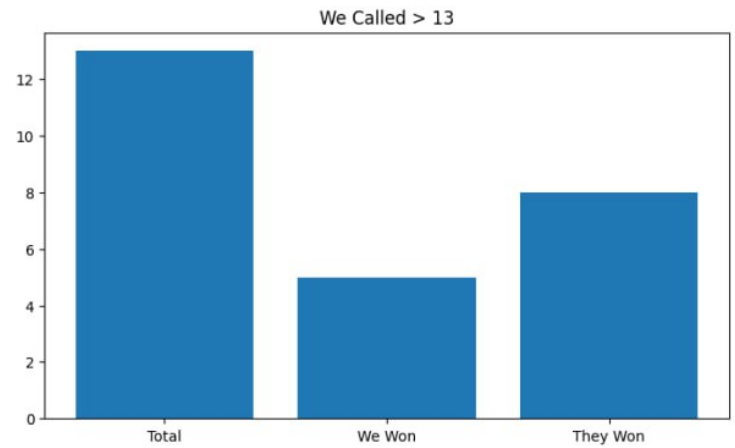
Combined Plot of the Three Versions

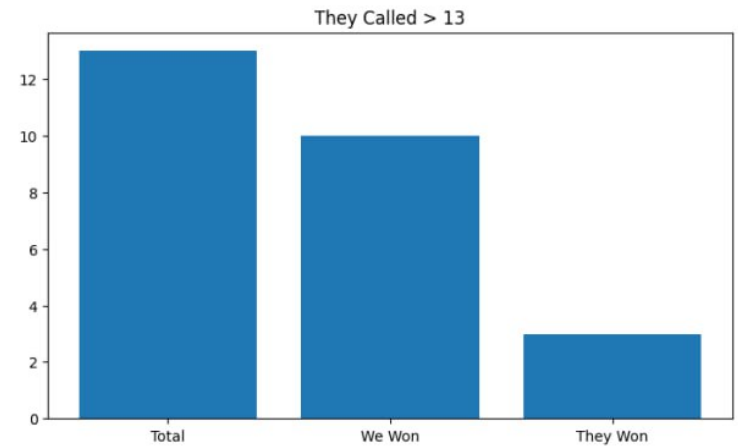


Game Results With Auction (Control)

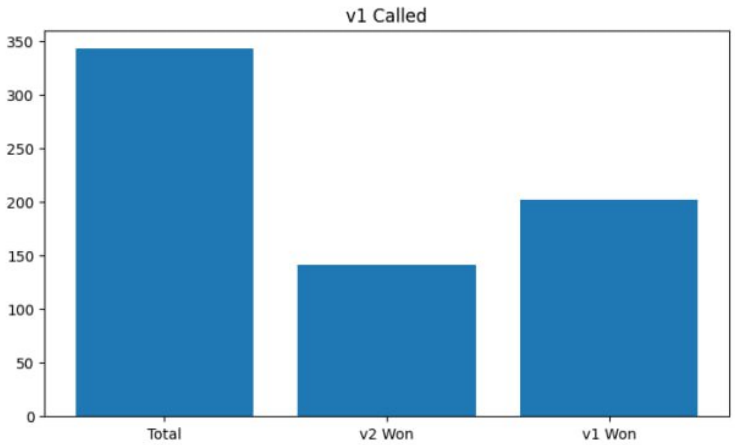


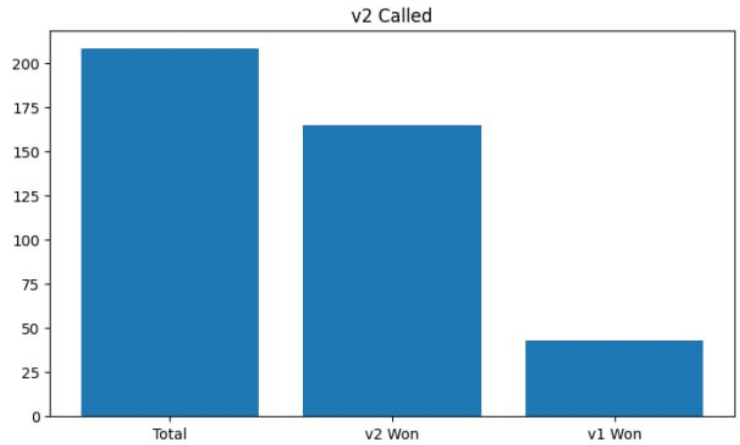


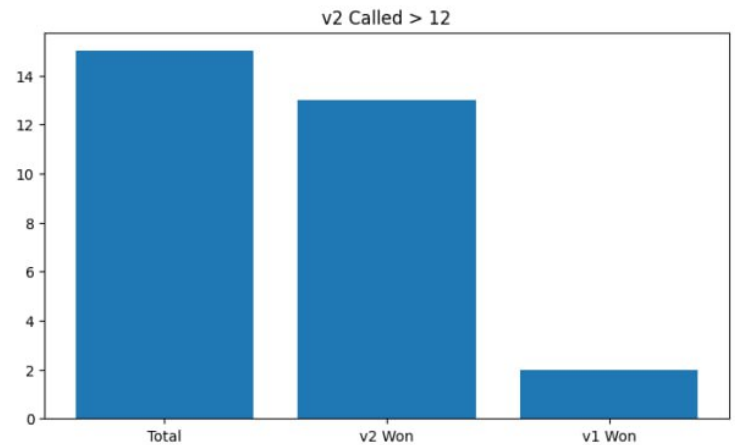


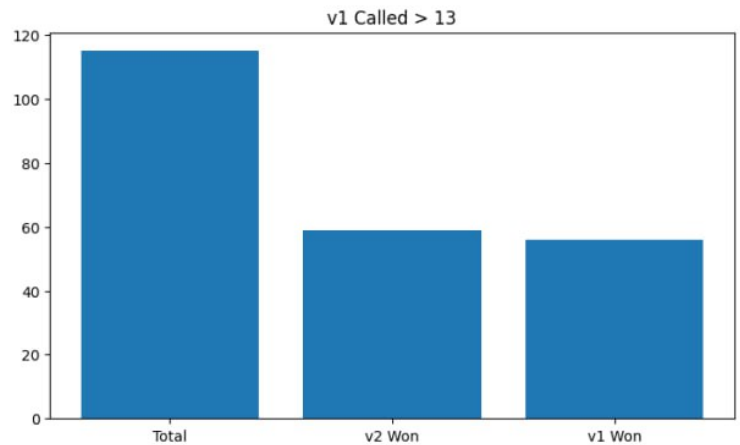


Game Results With Auction (V1 vs V2)









Numerical Solutions

- Euler's Method
- Backward Euler's Method

Euler's Method

By using the extension $y'(x) \approx \frac{y_{k+1}-y_k}{h}$ and having the y_0, we can iteratively predict the $y_{k+1} = y(x+h)$, by using the system

$$\begin{cases} y_{k+1} = y_k + h \cdot f(x_k, y_k) \\ y(x_0) = y_0 \end{cases}$$

Euler's Method

Euler's Method

In case of SIR, we are solving 3 ODE's, hence we are solving a system for s_{k+1}, i_{k+1}, and r_{k+1}, given the s_0, i_0, r_0

$$\begin{cases} s_{k+1} = s_k - \beta \cdot h \cdot s_k \cdot i_k \\ i_{k+1} = i_k + (\beta \cdot s_k \cdot i_k - \gamma \cdot i_k) \cdot h \\ r_{k+1} = r_k + \gamma \cdot h \cdot i_k \end{cases}$$

Backward Euler's Method

This time, the Method uses a different approach, namely

$$y'(x) \approx \frac{y_k - y_{k+1}}{h}$$

Plugging the SIR ODE's, we will get the system

$$\begin{cases} s_{t+1} = s_t - h \cdot \beta s_{t+1} i_{t+1} \\ i_{t+1} = i_t + h \cdot (\beta s_{t+1} i_{t+1} - \gamma i_{t+1}) \\ r_{t+1} = r_t + h \cdot \gamma i_{t+1} \end{cases}$$

By solving each one of these SLE's, we will get the next approximations for s_k, i_k, r_k

Backward Euler's Method

Simulations on USA's COVID-19 Data

Active Cases

Simulations on USA's COVID-19 Data

Simulations on Armenia's COVID-19 Data

Thank You For Attention!
More Information is available at Our Github Repository