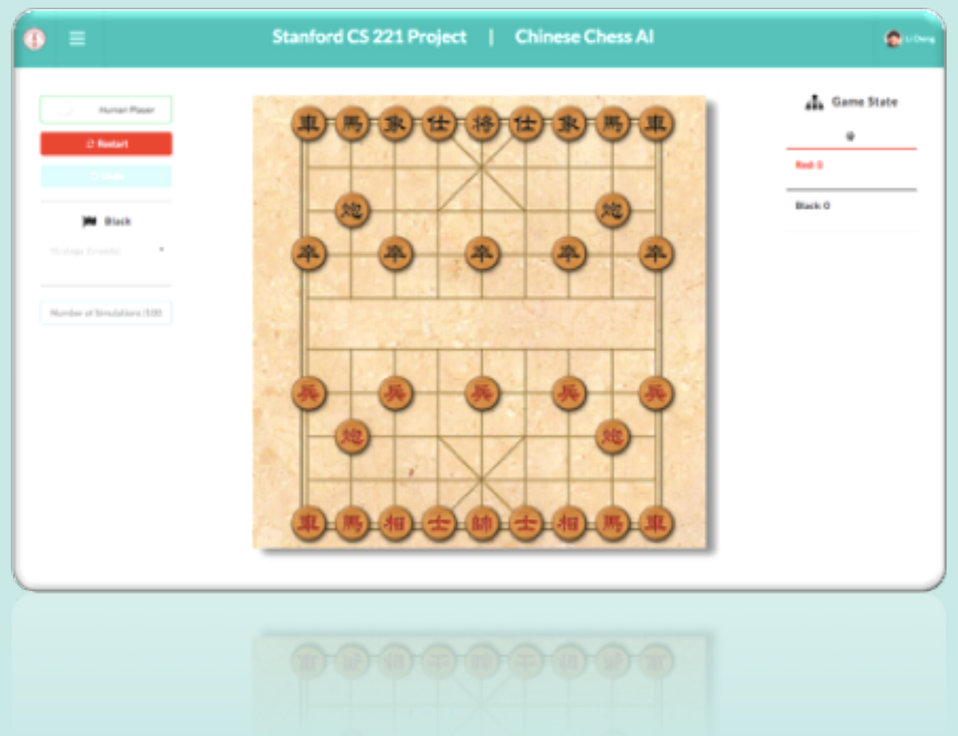
Chinese Chess AI

Simulator



- ❖ Human Mode
- ❖ Simulations Mode
- ❖ *Angular2 + SemanticUI*
- ❖ *Node.js*
- ❖ *Typescript*

Motivation

- ❖ Popularity in China
- ❖ Scarcity of AI-powered Game Engine

Challenge

- ❖ Complex Game Rules
- ❖ Large branching factor (38)

Evaluation Performance

Materials Value

- Piece Value
- Piece Position

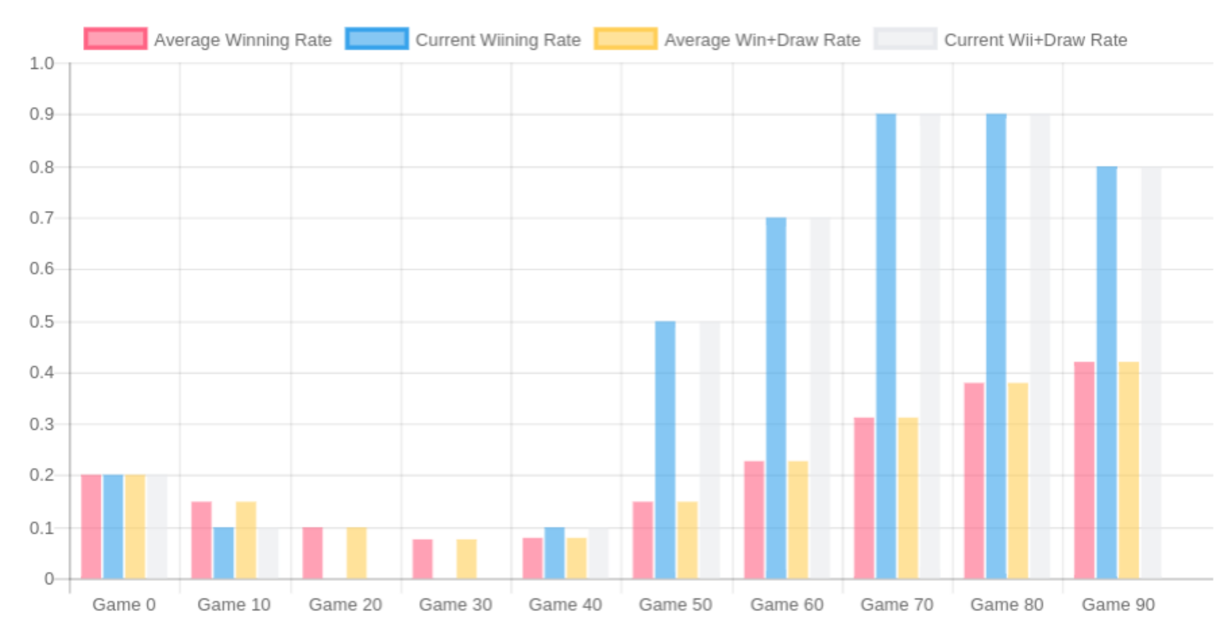
Attacking Power

- Number of Threatening
- Number of Captures
- Number of Center Cannons
- Number of Aligned Cannons

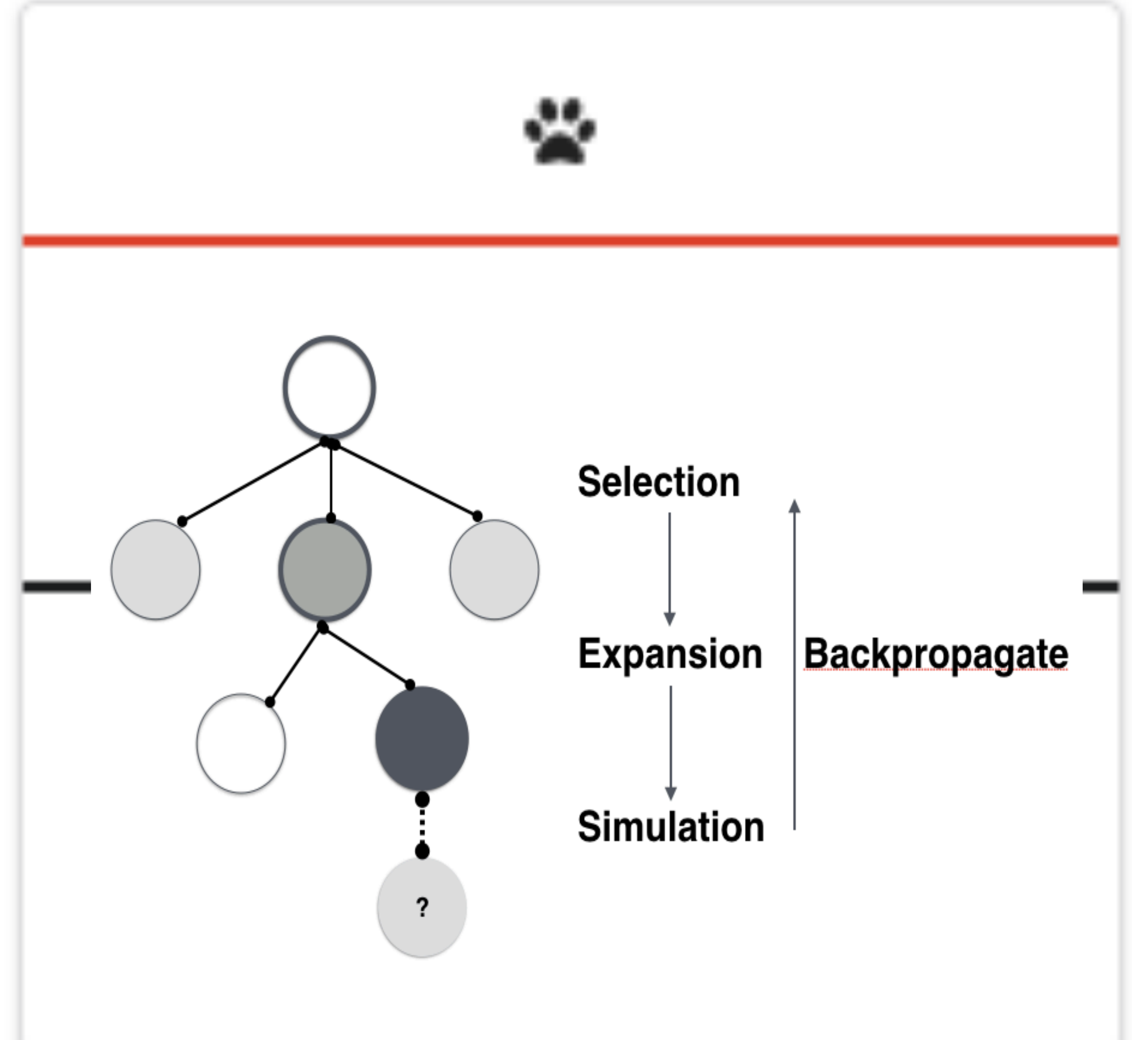
Mobility

- Mobility of Rook
- Mobility of Cannon
- Mobility of Horse
- Mobility of Elephant

Red Team: Wining Rate



Search Efficiency



Search Effenciency Comparison

Strategy	Search Depth	Average Runtime for Each Move(ms)
Alpha-Beta Pruning	2	76
Greedy	1	3
Alpha-Beta Pruning	3	600
Alpha-Beta Pruning	4	7307
Alpha-Beta Pruning with Move Reorder (Type A)	2	72
Alpha-Beta Pruning with Move Reorder (Type A)	3	239
Alpha-Beta Pruning with Move Reorder (Type A)	4	3175
Temporal Difference Learning	2	393
Temporal Difference Learning	3	1176
Temporal Difference Learning	4	9568
Monte Carlo Tree Search	2	43
Monte Carlo Tree Search	3	105
Monte Carlo Tree Search	4	315

$$w^{t+1} := w^t + \eta \cdot f(\sum \phi_i) \times r^t$$

ϕ_i: feature vector saved during game t for state i

f: normalizing function $f = (\frac{1}{1+\exp^{-x}} - 0.5) \times A$

r^t: reward at game $r = \begin{cases} Win: 1 \\ Lose: -1 \\ Draw: 0 \end{cases}$

A: scaling parameter

Strategy

- ❖ Greedy
- ❖ MiniMax
- ❖ Alpha-beta Pruning
- ❖ Pruning with Move Reorder
- ❖ Monte Carlo Tree Search
- ❖ Temporal Difference Learning

Github

Li Deng
dengl11@stanford.edu
Stanford CS221 2016 Autumn