Go and AI

Outline

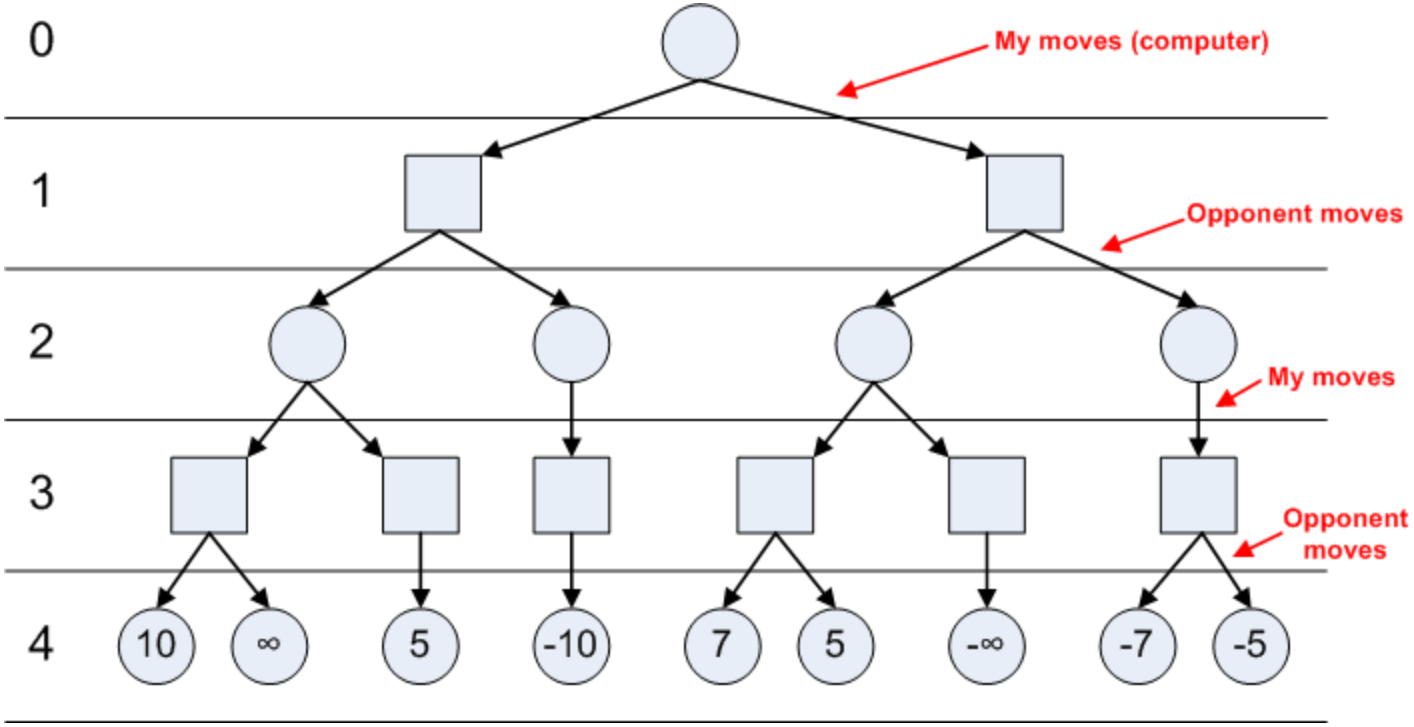
- Brief explanation of traditional gaming AI
- Introduction to Go
- Why usual techniques utterly fail for Go
- So what's the answer?

- NOT A philosophical discussion about what intelligence is!

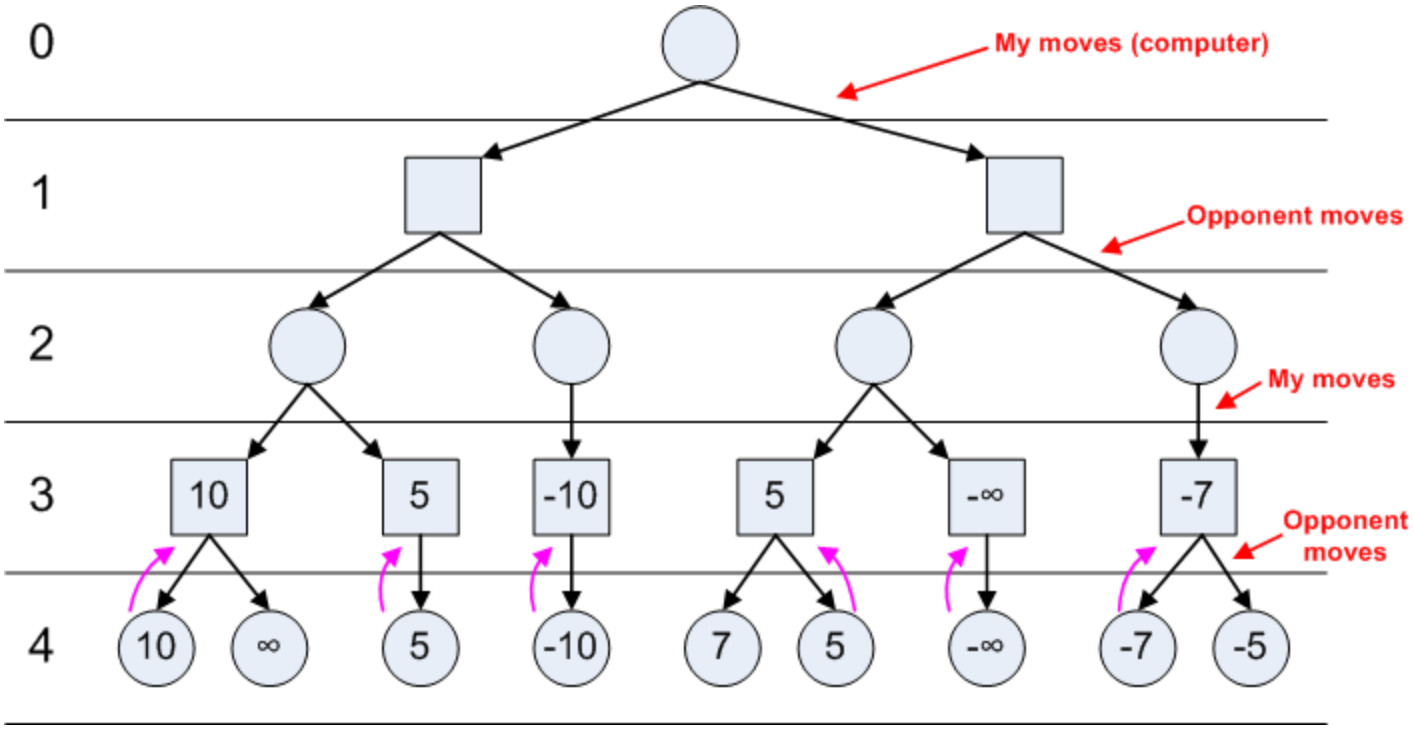
Traditional AI

- Mainstay of traditional AI techniques is ‘minimax’ algorithm
- Outline
  - looking ahead x number of moves, for every branch of every move
  - Applying evaluation function to score the position
  - Choose move based on best score x moves ahead
  - If it is your move, maximize score
  - If it is opponent’s move, minimize score

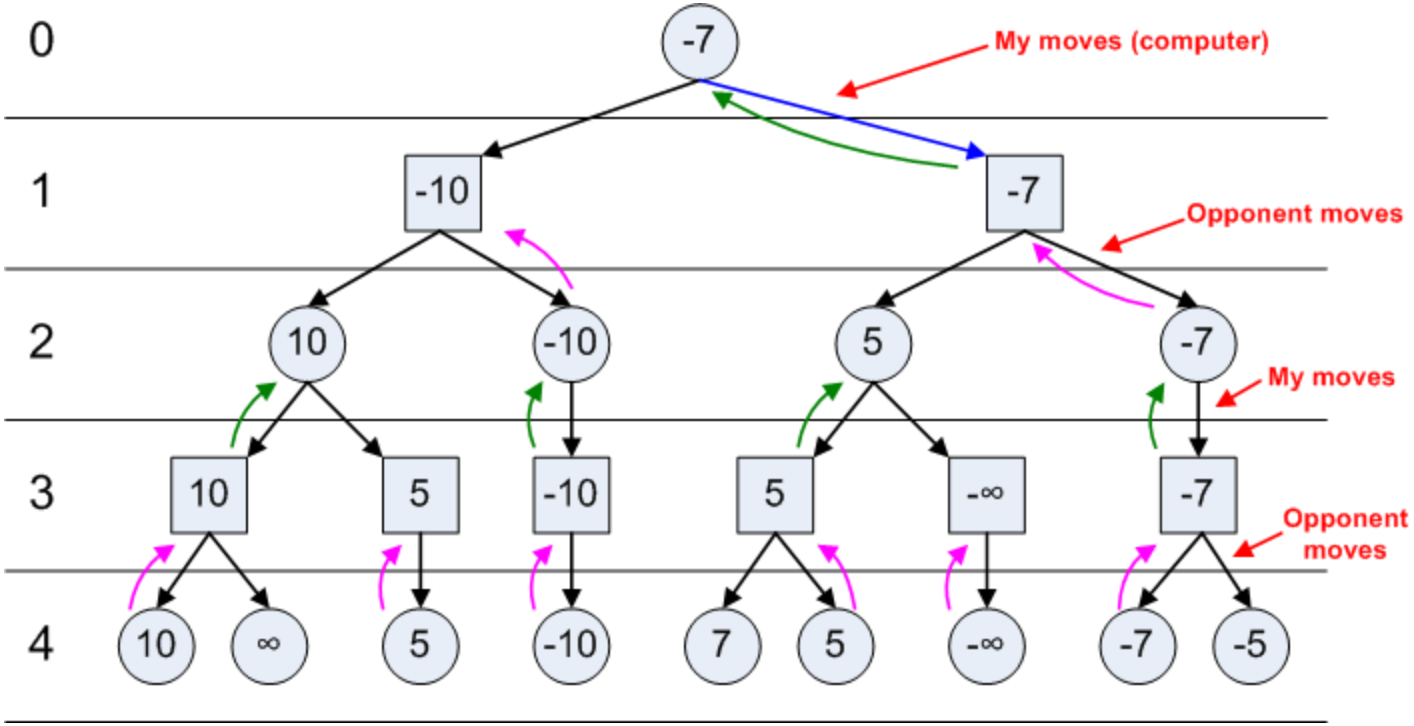
Minimax


0
1
2
3
4
My moves (computer)
Opponent moves
My moves
Opponent moves
10
∞
5
-10
7
5
-∞
-7
-5

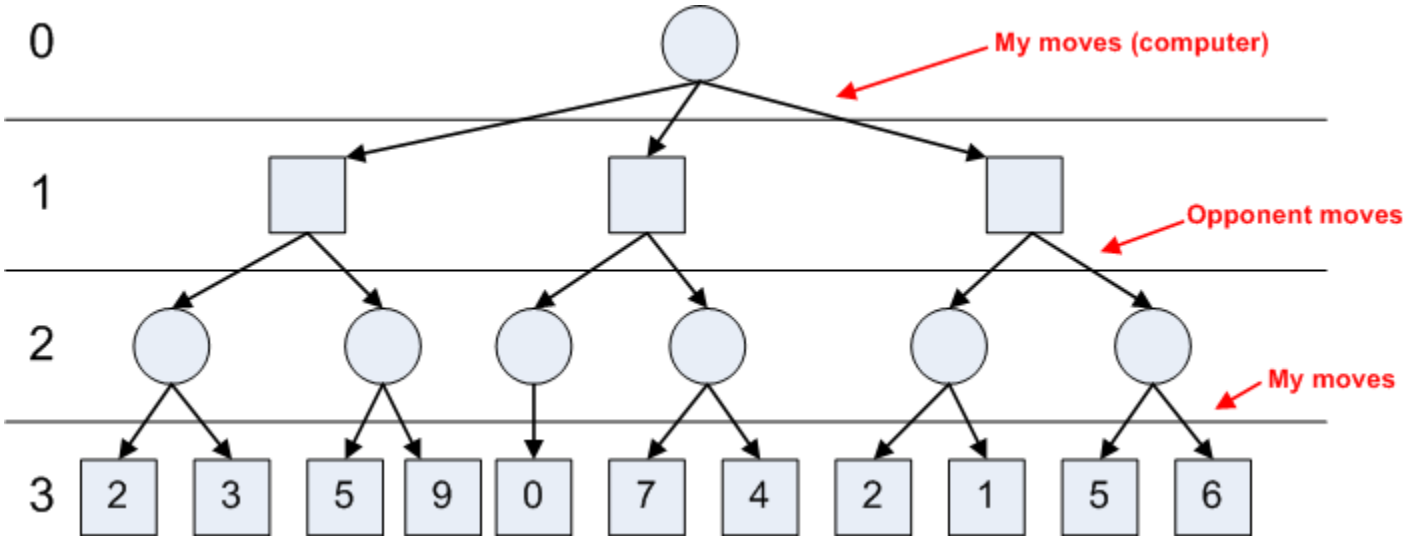
Minimax


0
1
2
3
4
My moves (computer)
Opponent moves
My moves
Opponent moves
10
5
-10
5
-∞
-7
10
∞
5
-10
7
5
-∞
-7
-5

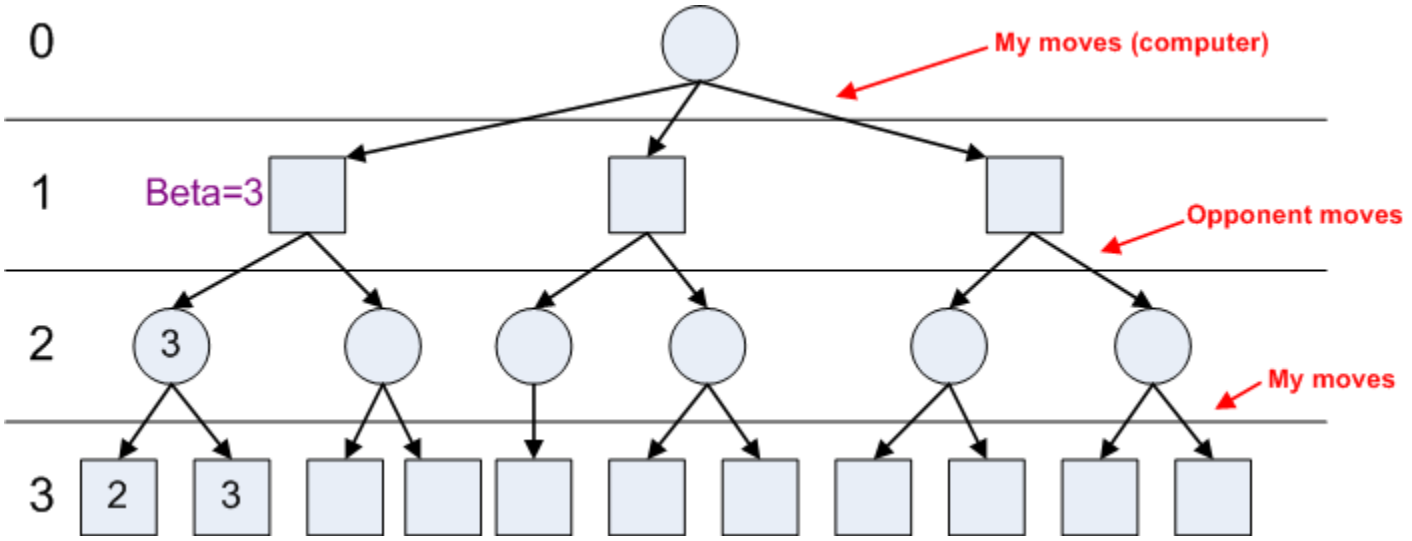
Minimax


0
1
2
3
4
-7
My moves (computer)
-10
-7
Opponent moves
10
-10
5
-7
My moves
10
5
-10
5
-∞
-7
Opponent moves
10
∞
5
-10
7
5
-∞
-7
-5

Pruning – Alpha Beta

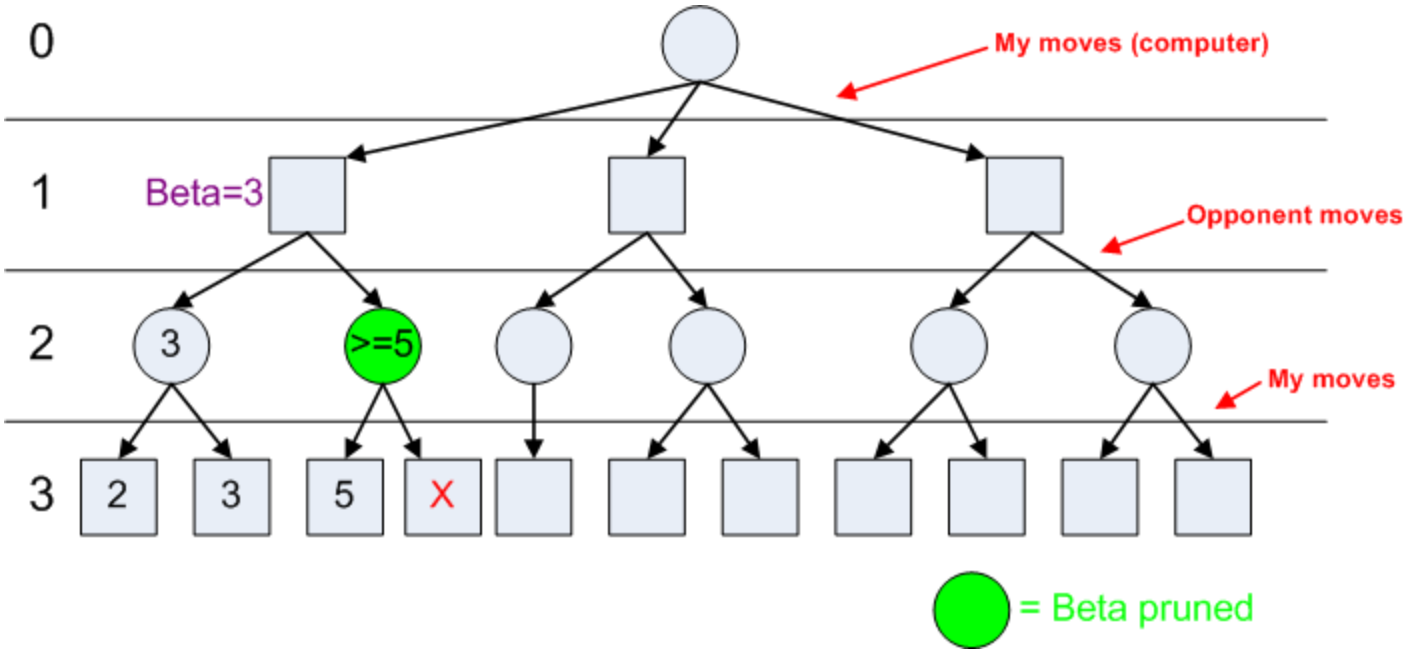


Pruning – Alpha Beta

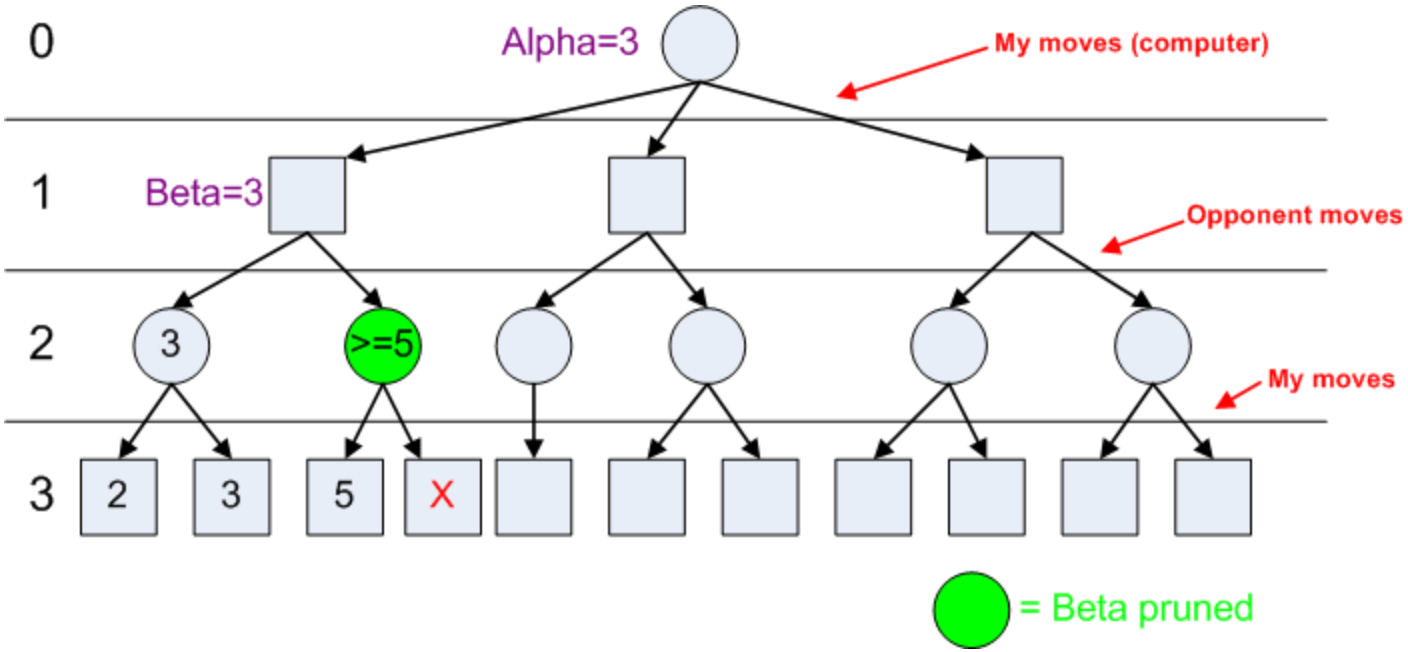


- Beta – node will never exceed this value (since we are minimizing)

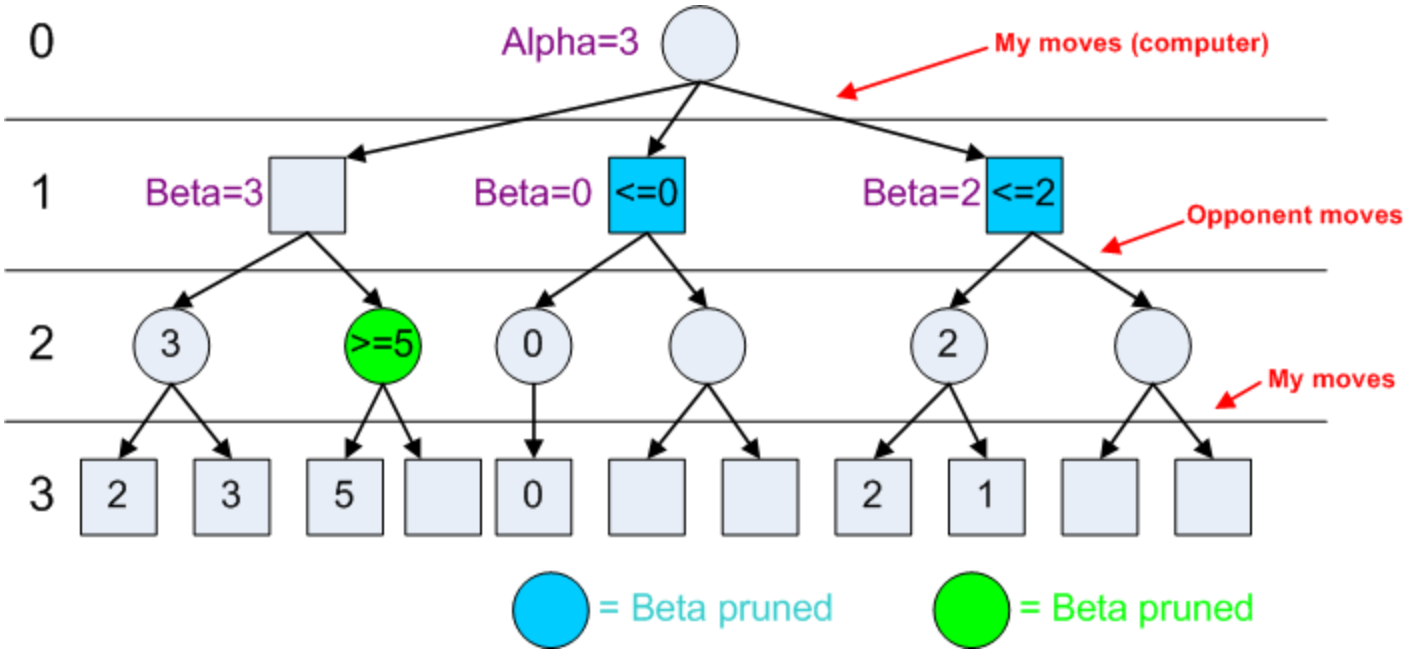
Pruning – Alpha Beta



Pruning – Alpha Beta



Pruning – Alpha Beta



Evaluation Function

- Calculate various aspects of the board, and give each aspect a weight
- c_1 * material + c_2 * mobility + c_3 * king safety + c_4 * center control + ...
- Chess computer programs tend to give material advantage a big weight (favouring tactics over strategy)

Pruning – Transposition Tables

- Simple idea – multiple different moves may result in similar game state
- Keep a record of game state scores
- Zobrist hashing

Deep Blue

- 1997 – first program to beat grandmaster in 6 game match using standard time controls
- 30 processor computer, enhanced with 480 special purpose chess chips
- Ran on AIX platform ☺
- 200 million positions evaluated per second
- 8 moves ahead (but up to 20)
- Key – accurate and complex board evaluation

Modern Chess Software

- Deep Fritz, Rybbka
- Usually run on PC's (dual core)
- Substantially more efficient than Deep Blue
- Only 8 million positions per second
- But look ahead 18 moves on average

Is this AI?

- Minimax+pruning effective for chess and checkers, which have small branching factor
- Something of a fluke that humans consider chess interesting
- With minimax, impossible to explain to human why the move is “good”
- Relies on accurate evaluation function
- **Does not readily extend to other domains**

What is Go

- History
- Concepts
- Rules
- Basic Tactics and Strategy

Go the board game

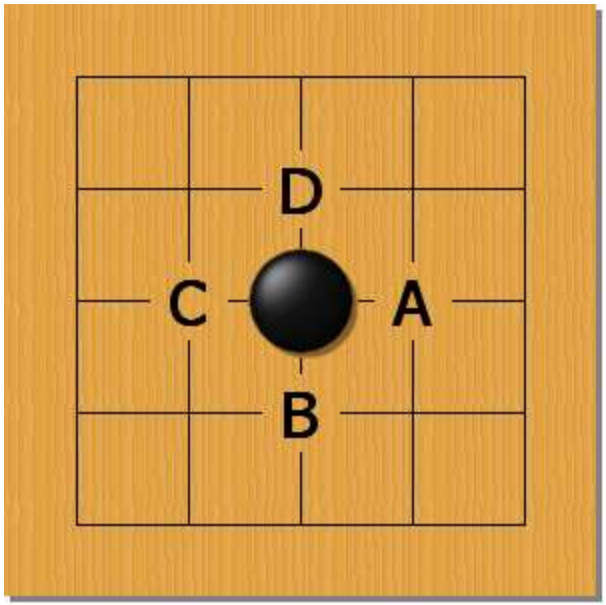
- About 3000 years old – the oldest board game
- Very simple rules, very high complexity
- Very popular in China, Japan, Korea

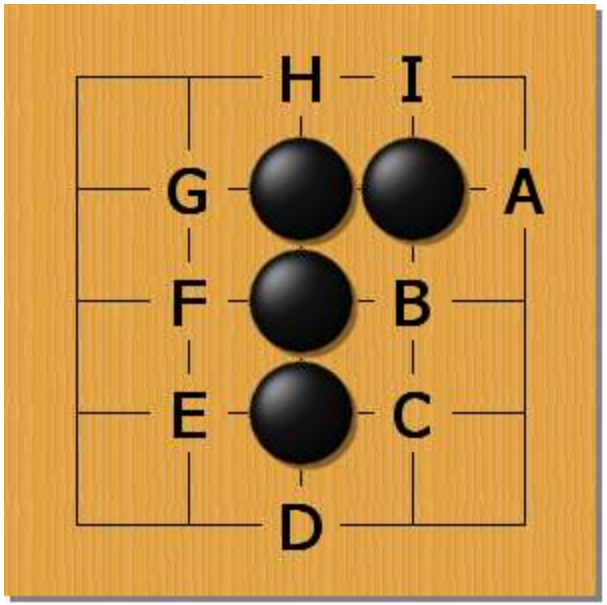
Go Concepts

- Played on a square board – with 19x19 intersections
- Smaller board variations played too, e.g. 9x9
- Black starts, and players alternate in placing stones on intersections
- Stones never move once played, but if captured, they get removed from the board
- Objective – capture enemy stones, and surround territory
- White gets point compensation for going second, usually 6.5 points (prevent draws)

Capturing stones

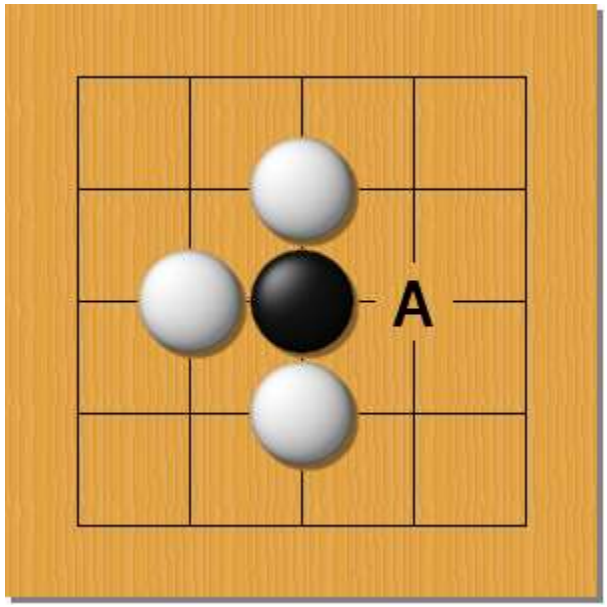
- Liberties

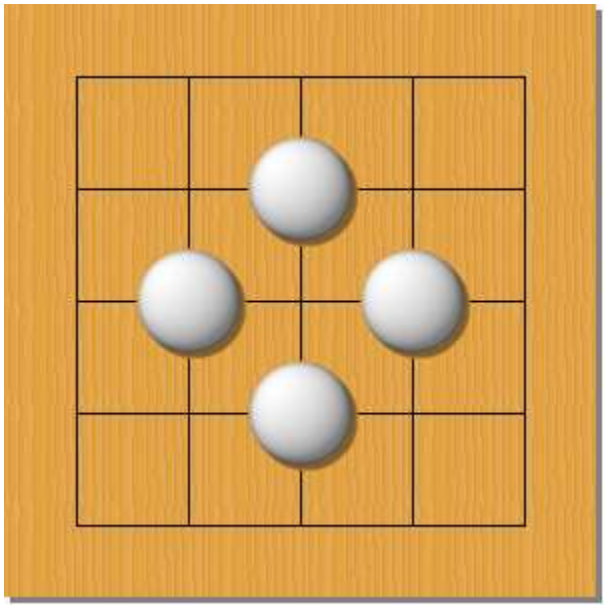




Capturing stones

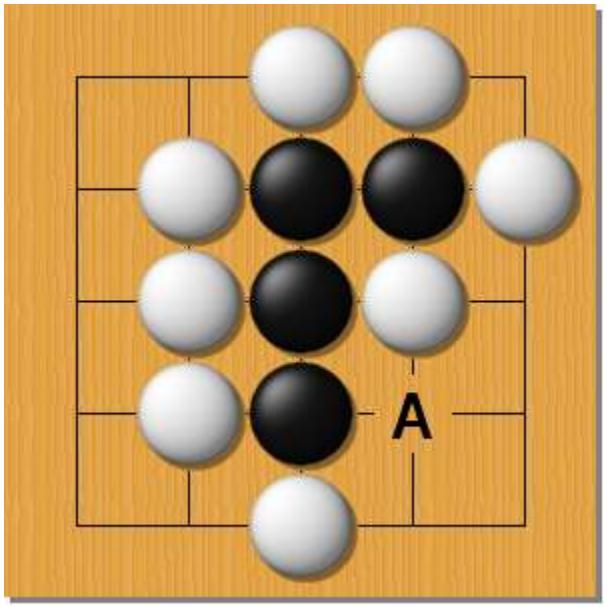
- Removing liberties

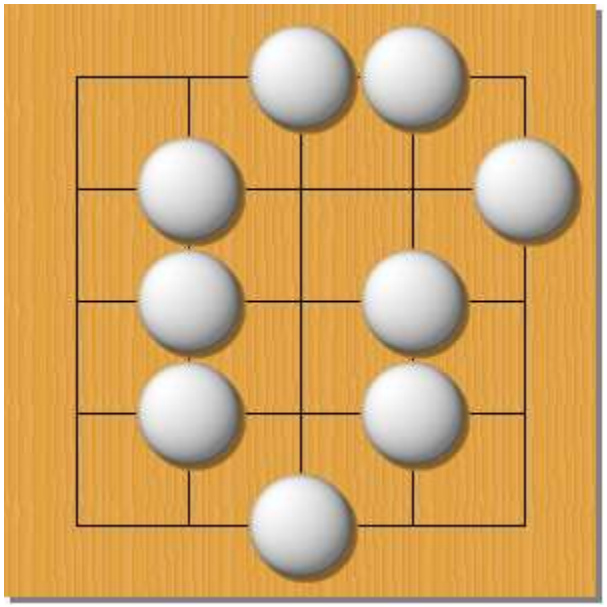




Capturing stones

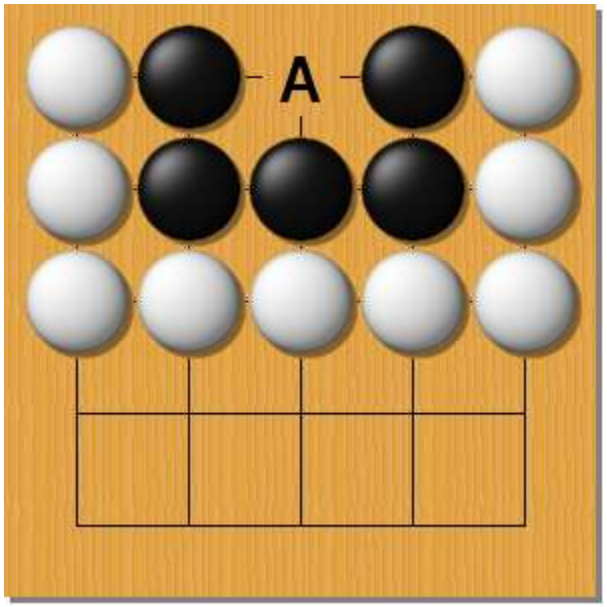
- Removing liberties

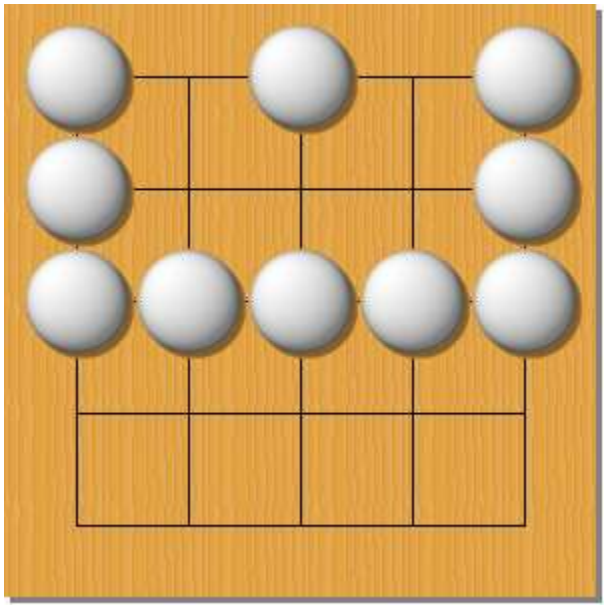




Capturing stones

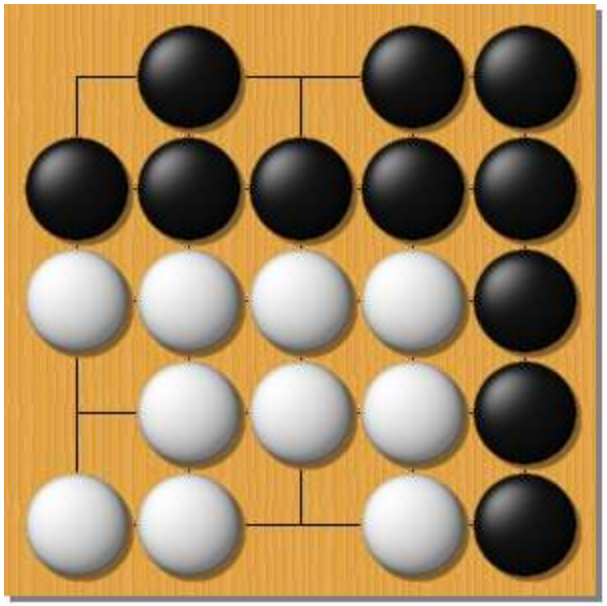
- Suicide illegal, unless capturing stones

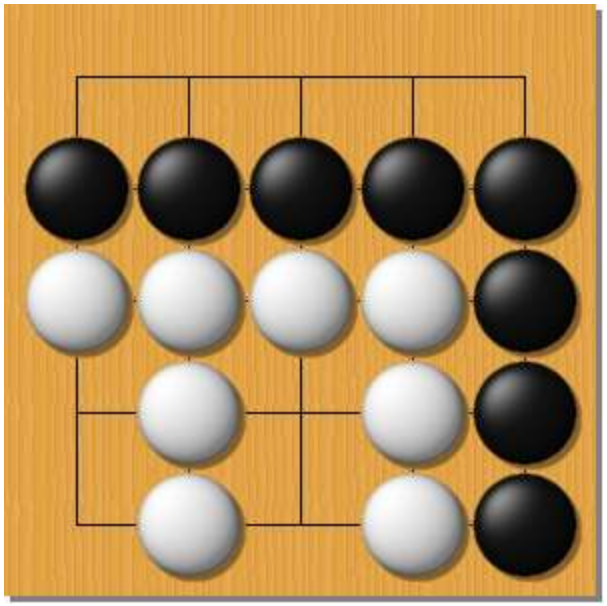




Life

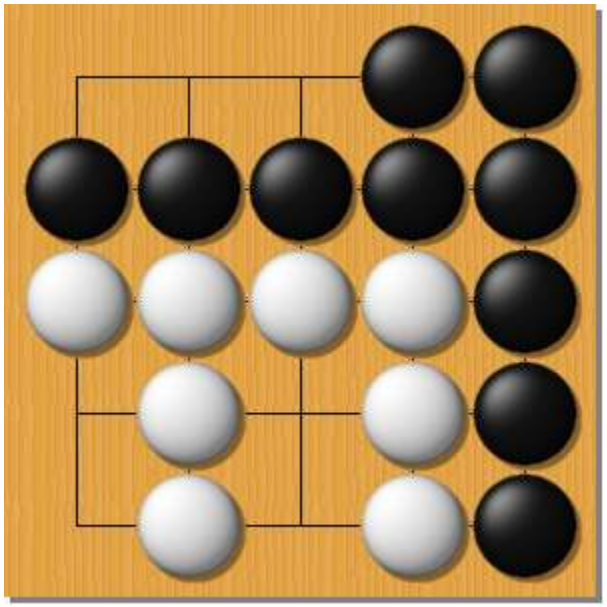
- Out of capture and suicide rules, arises fundamental principle of Go – two eyes

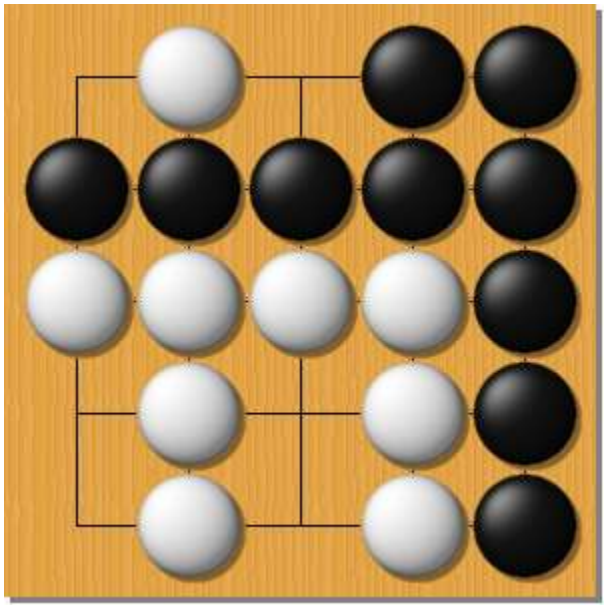




Death

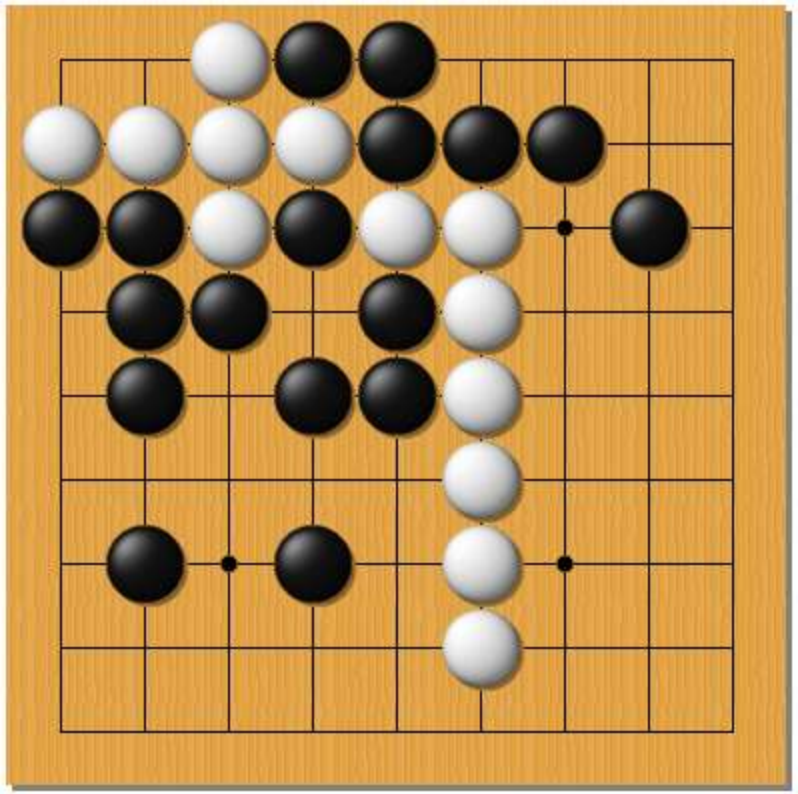
- Kill opponents by preventing two eyes

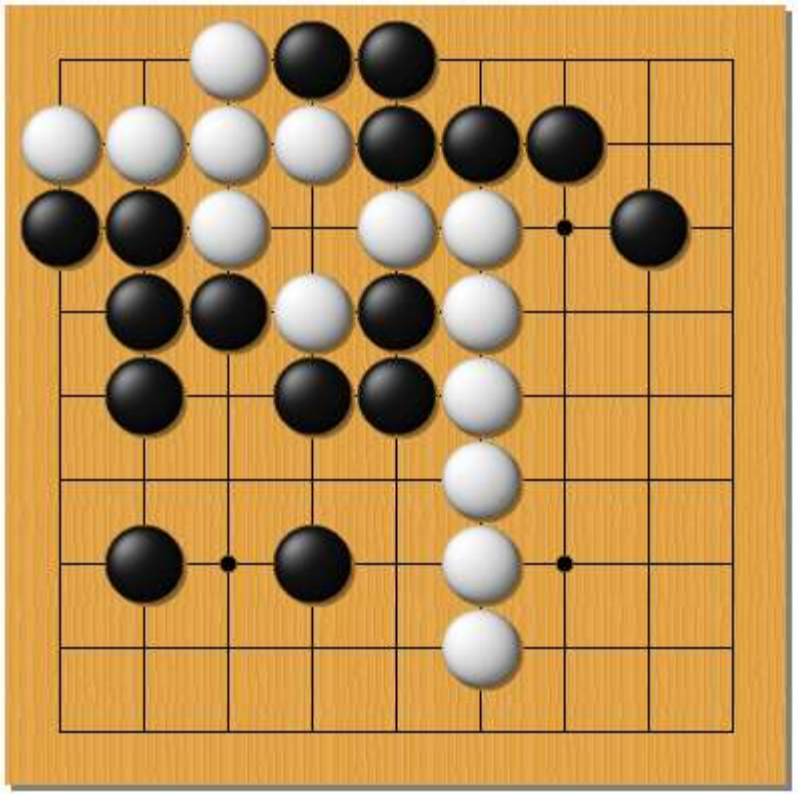




Ko

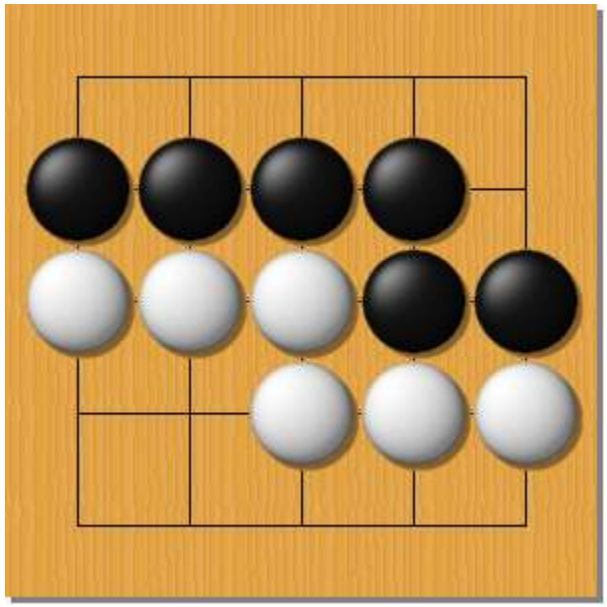
- May not repeat a board situation



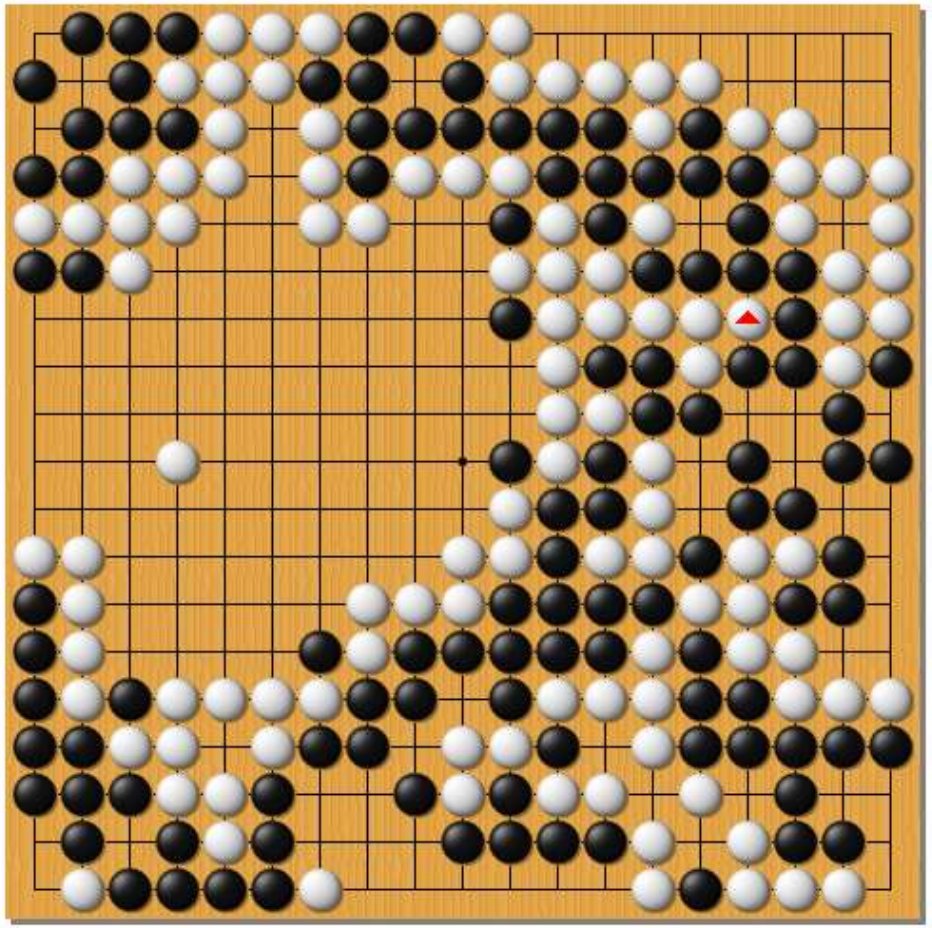


Territory

- Object is gain more territory than opponent

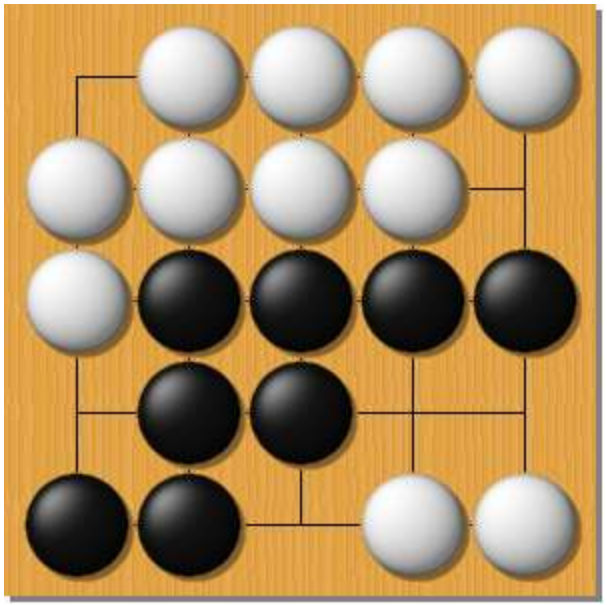


Territory



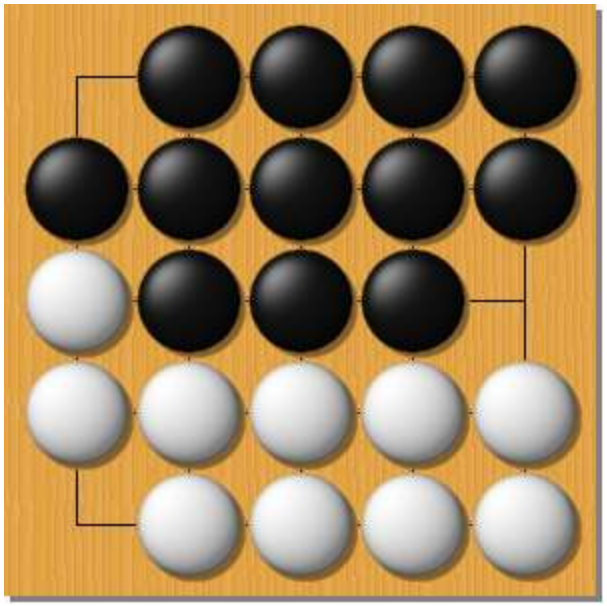
Tactics – Race

- Capturing Race – when two opposing groups don't have two eyes, and are trying to capture each other – one with most liberties wins



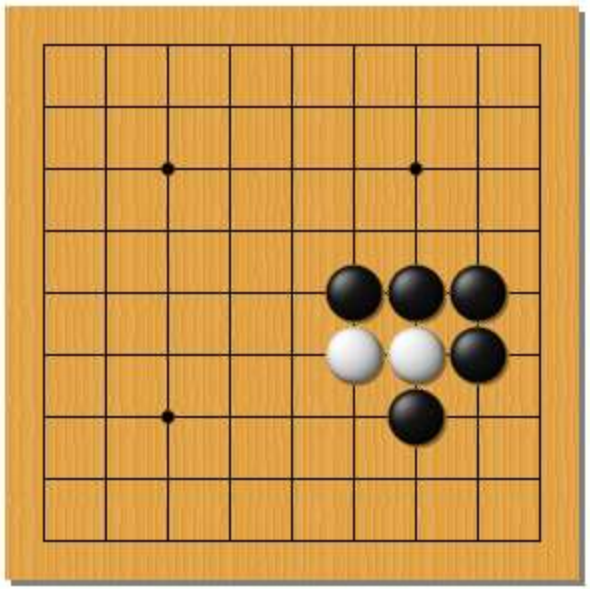
Tactics – Seki

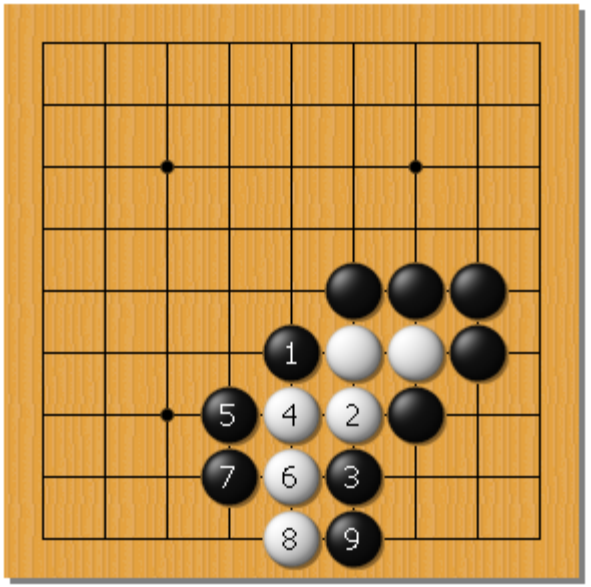
- Two opposing groups do not have two eyes, but neither can capture each other



Tactics

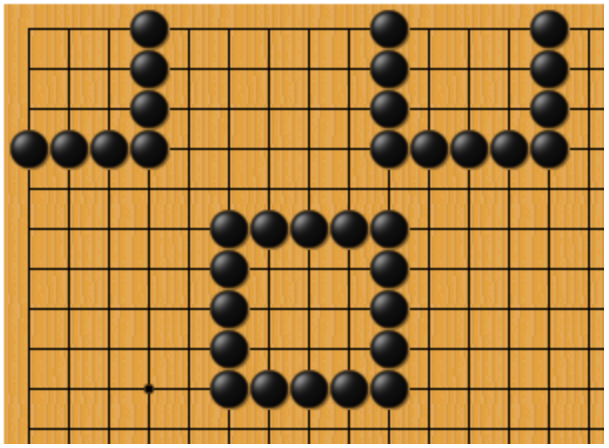
- Ladder – player keeps escaping, but opponent keeps putting it in atari, and eventually captures. Many moves ahead, but easy to read, even for beginner



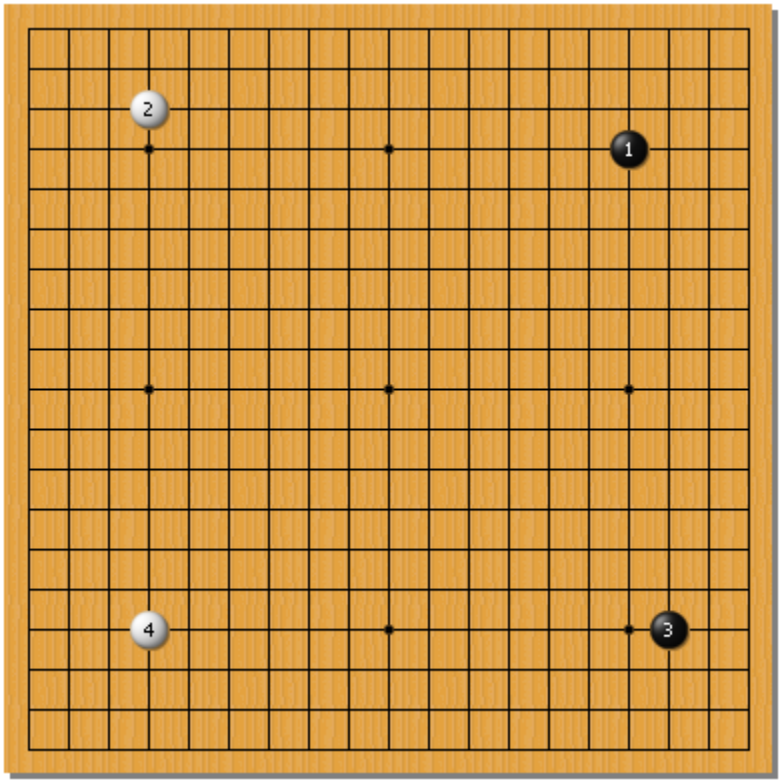


Strategy – Territory

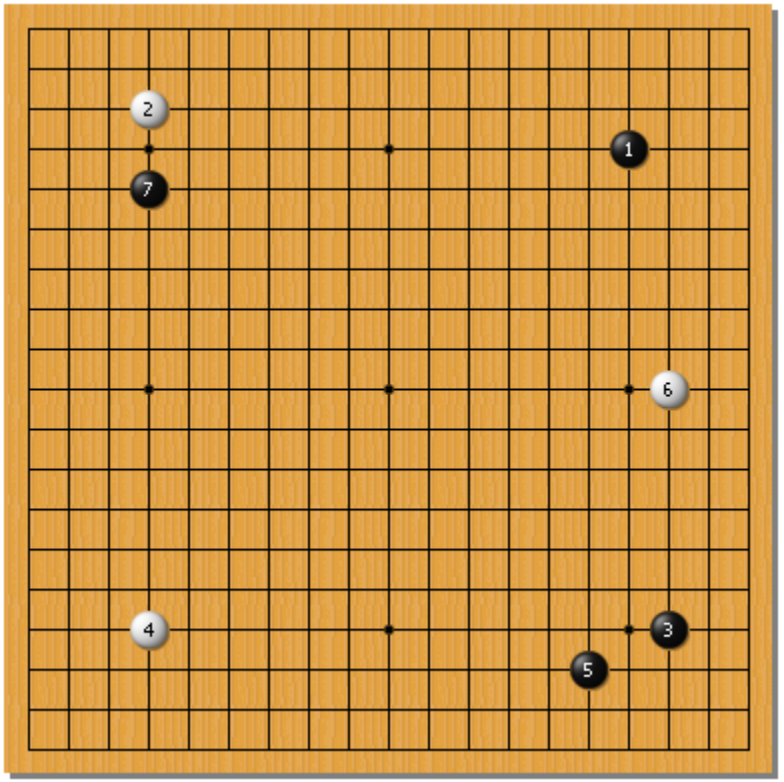
- First corners
- Then sides
- Finally center



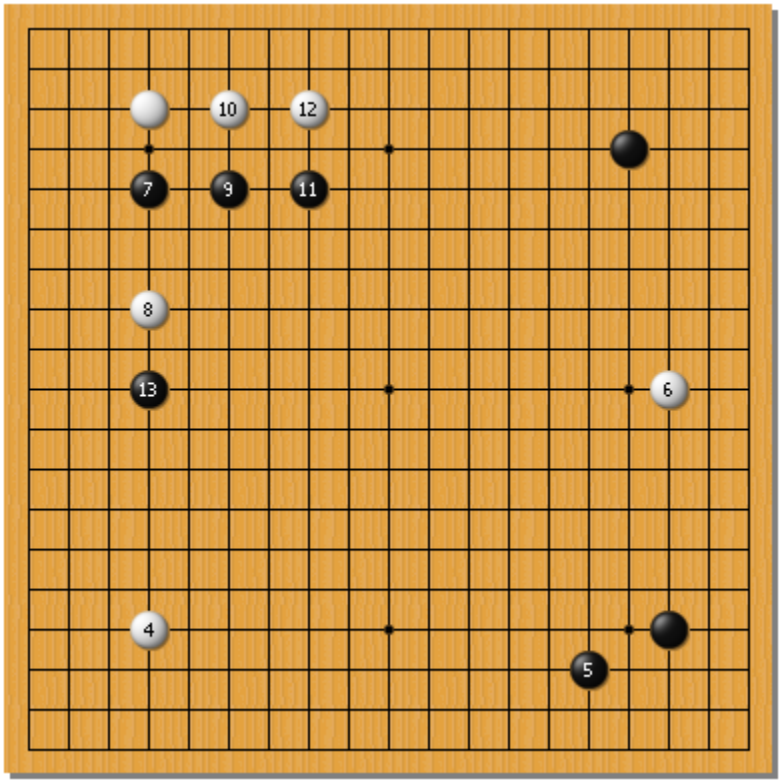
Strategy – Opening



Strategy – Opening

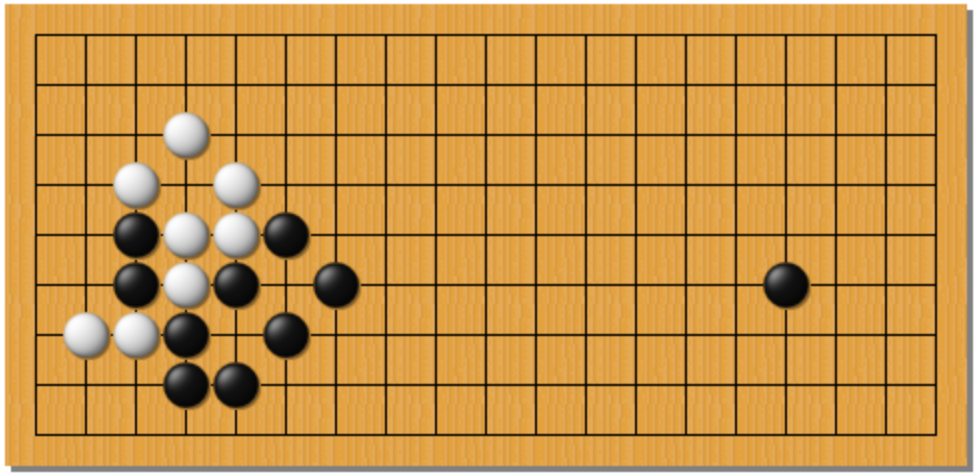


Strategy – Opening

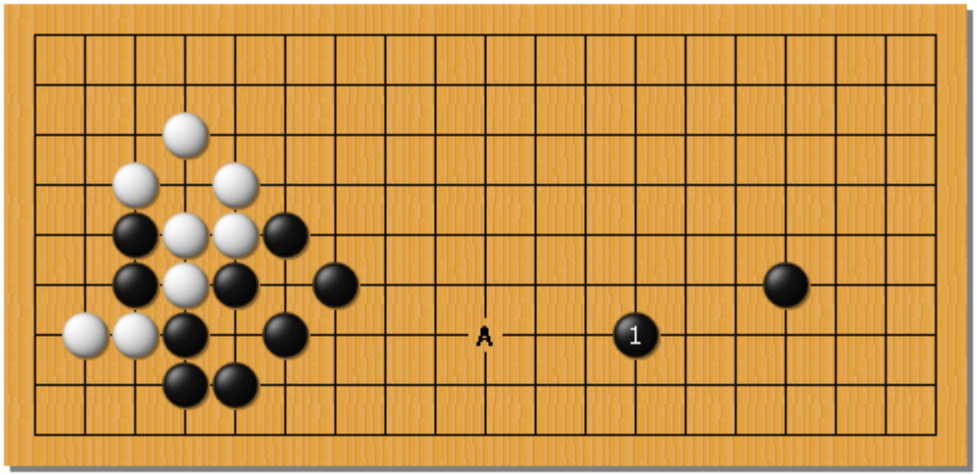


Strategy – Efficiency/Thickness

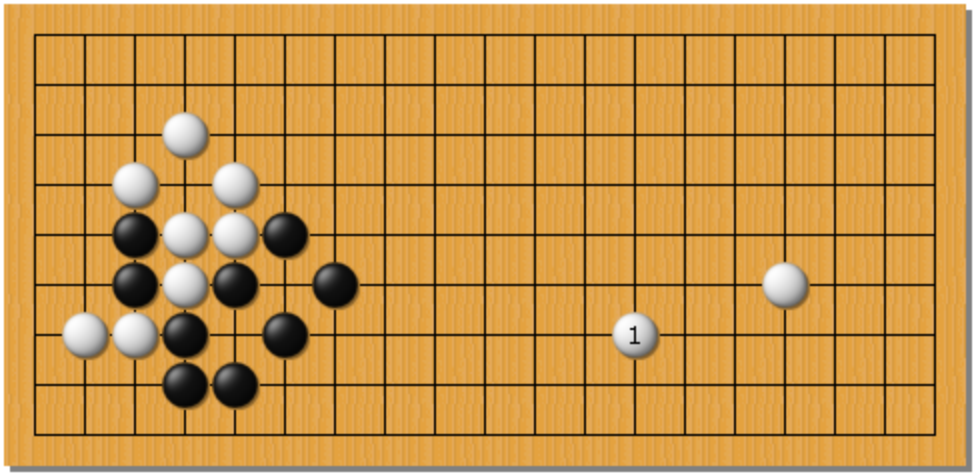
- Surround territory efficiently
- Don’t play too close to strong stones



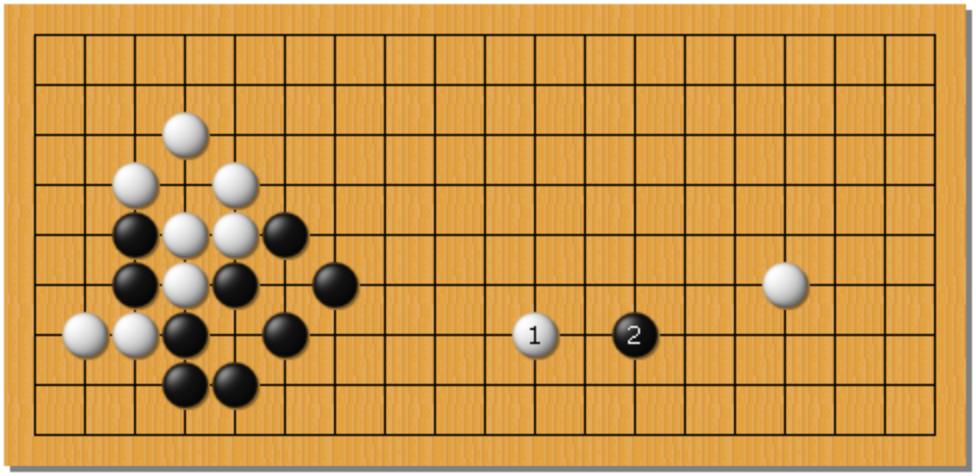
Strategy- Efficiency/Thickness



Strategy- Efficiency/Thickness

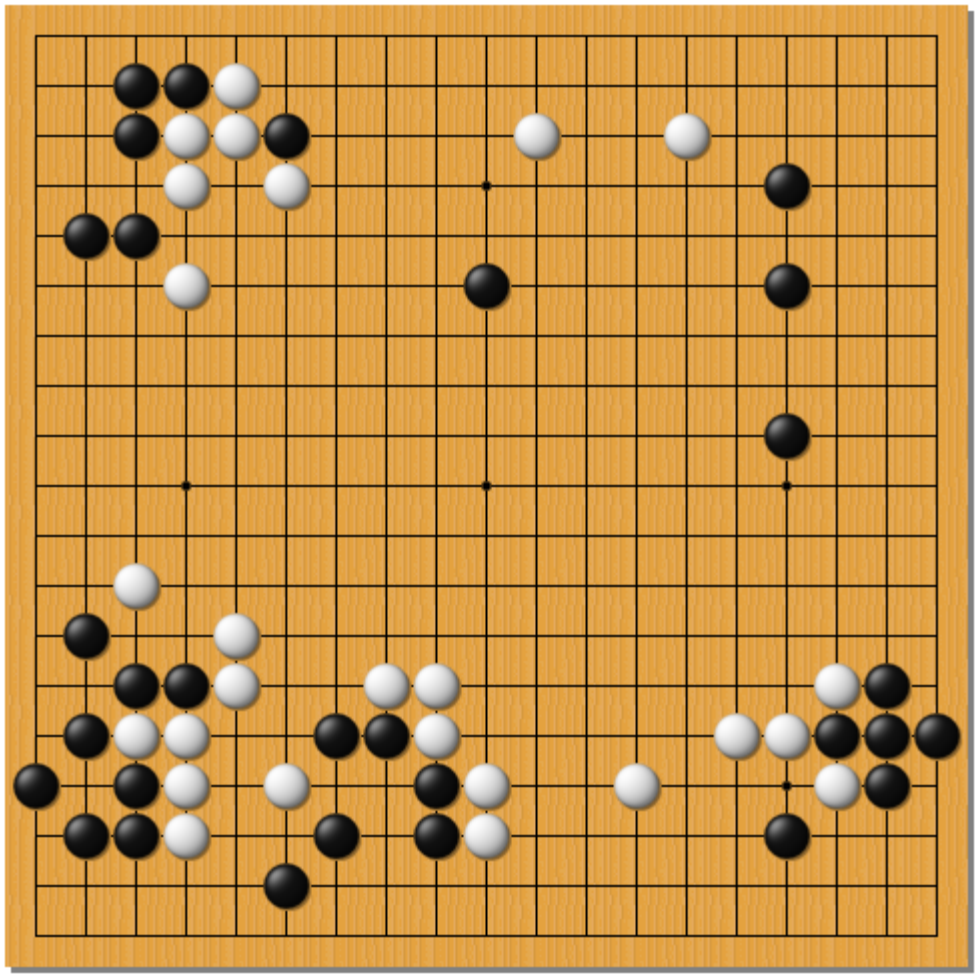


Strategy- Efficiency/Thickness

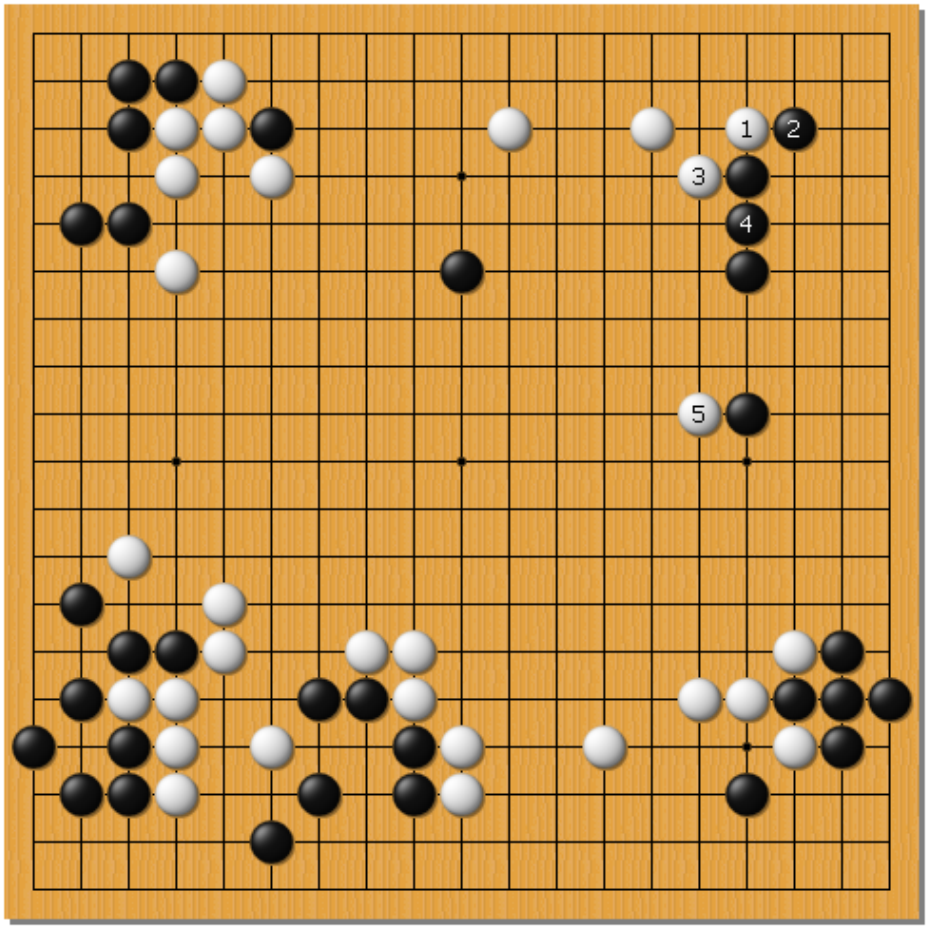


Strategy – Leaning Attack

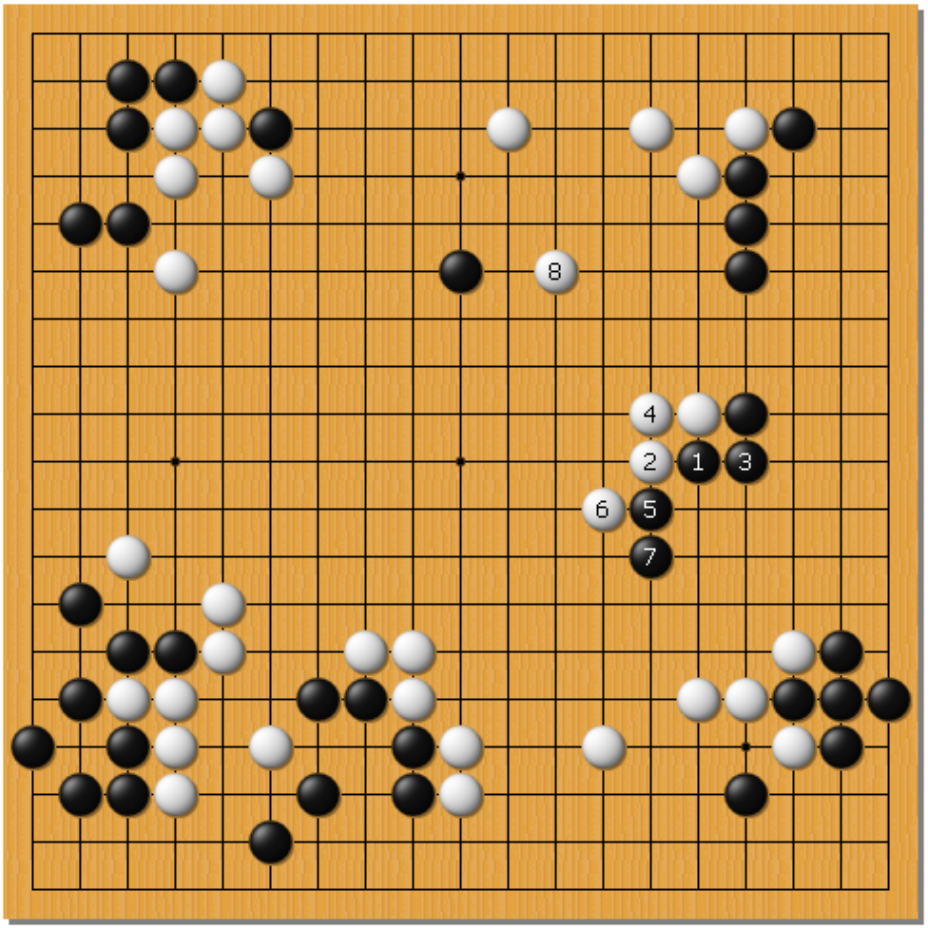
- Make a feint to the east while attacking in the west



Strategy – Leaning Attack



Strategy – Leaning Attack



Principles

- Go is all about efficiency
- Go is about cooperation
- Pattern recognition and intuition important

- But just so it's clear
  - Go is a fighting game
  - Reading ahead is still the most important skill

Ranking

- 30 kyu … absolute beginner)
- 15kyu … average beginner after 2 months
- 5kyu – 1kyu … intermediate
- 1dan – 9dan … expert to expert amateur
- 1pro – 9pro … professional ranks

- Means there are about 40-45 distinct skill levels (chess has about 18)
- Handicapping very effective – 1 stone head start per difference in rank

Solving Go – Why so hard?

- Extreme game tree complexity (number of minimax nodes to evaluate initial position)
  - Chess – 10^{126}
  - Go – 10^{360}
- Extreme Branching
  - Chess – +/–35
  - Go – 300
- Evaluation Function extremely difficult
  - Has to take into account changing status of stones
  - E.g. locally dead groups may become alive after ko fight
  - Strategic concepts very hard to quantify
  - E.g. influence, thickness, extent to which weak groups can be attacked
- Pruning Branches very hard

How is it done

- Alpha, Beta Search + hand tuning
  - ~6kyu
- Neural Networks
  - ~15kyu
- Monte Carlo (+Tree Search)
  - ~2dan

Why are humans so good?

- In contrast to many other games, Go is additive
  - Stones get added to the board, and rarely removed
  - Makes it much easier to read far ahead, in contrast to chess or reversi

Pattern recognition very useful in Go

E.g. recognizing strong or weak shapes

Why classic techniques fail

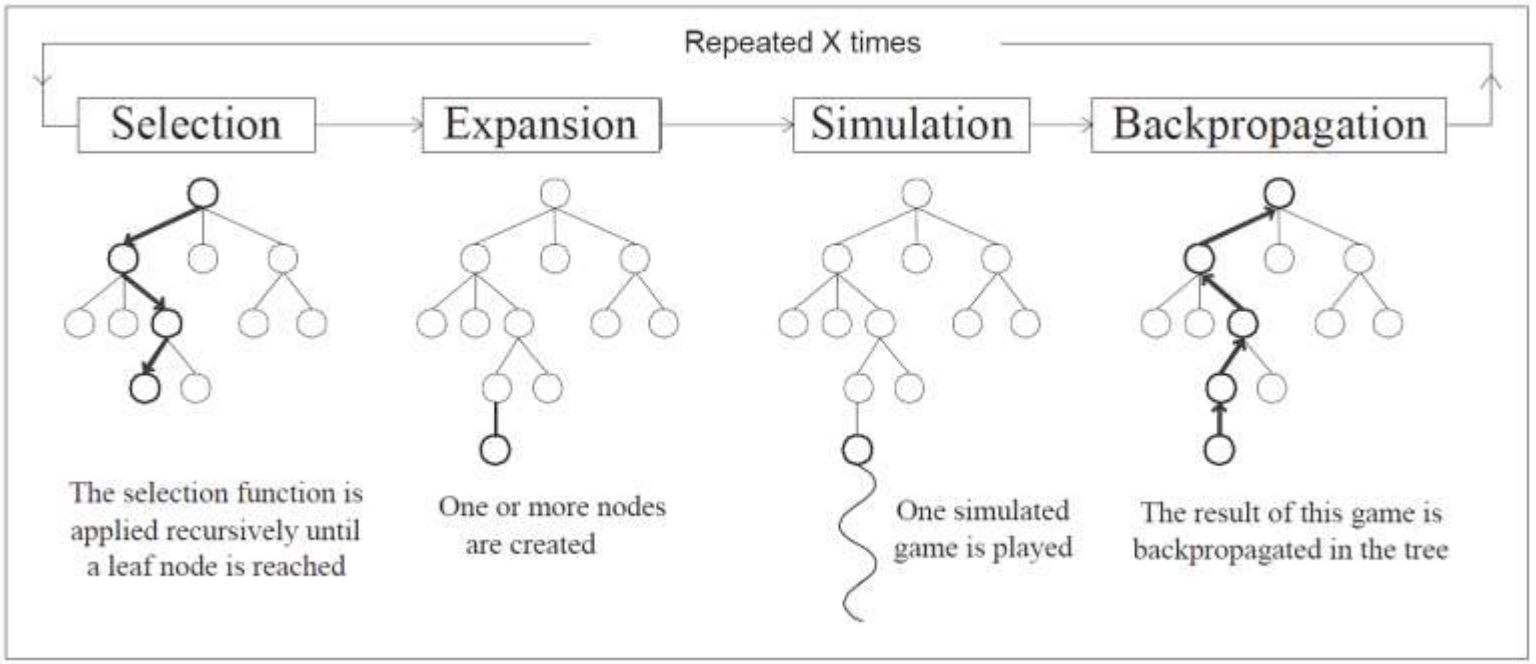
- Frequently miss critical moves – overpruning
- Drastic misjudgements of group status
- Unable to adjust style for different situations
- Strength does not scale with thinking time

Monte Carlo

- Random playouts – play out thousands (or millions) of games, randomly
- Pick move with highest winning rate
- Remarkably, this incredibly simple approach can reach around 5k
- True breakthrough came in 2006, with Monte Carlo Tree Search

MCTS (or MC/UCT)

- Structured format for playouts



- Selection/Simulation key areas of research

MCTS

- Selection must balance exploitation and exploration
- Exploitation – revisiting nodes that have high winning rates
- exploration – finding new nodes that may have better results

- Simulation – light playouts vs heavy playouts
- Light – (almost) no knowledge
- Heavy – applying knowledge (tactical)

Successes of MCTS

- 2008 first time pro beaten at 19x19 with 9 stone handicap
- 2008 first time pro players beaten at 9x9
- 2009 first time pro beaten at 19x19 with 6 stone handicap
- Bots on go servers have reached 2 dan
- Possibly 3 dan in special events (many nodes)

Application of MCTS

- Readily extends to other domains
- E.g. general planning tasks with large search space and stochastic evaluation function
- E.g. other games (Poker, Arima, ...)

Aesthetic Problems

- Computers tend to play strange looking moves, like in center in the opening
- Play sub-optimally near the end when win is assured (irritating)
- Play ridiculous moves when position is lost (they have a non-zero chance of success, but irritate experience players)

Is there a future for MCTS and Go?

- Hottest area of research at the moment
- Stb university have open bursaries for masters students to research MCTS
- MCTS will eventually search every possible tree – but resources will exceed alpha-beta
- Despite apparently lending itself well to multi-cores, it's not scaling well
- Back to where we were before MCTS (clever use of heuristics to narrow search space)

Applying Strategic Concepts

- Attractive, as simulations cannot be deep enough to assess true value of strategic concepts
- E.g. thickness, ko fight status, latent potential
- No published results of anyone achieving this yet

Credits

- Bulk of talk based heavily on a presentation by Pasky http://pasky.or.cz/~pasky/go/
- Front page diagram from Wikipedia
- Game positions created with MultiGo http://www.ruijiang.com/multigo/
- Minimax diagrams created with Visio
- MCTS diagram copied from dissertation **"Monte-Carlo Tree Search: A New Framework for Game AI" by Guillaume Chaslot, Sander Bakkes, Istvan Szita and Pieter Spronck**