More on Games

Chapter 5.4-5.7

Some material adopted from notes by Charles R. Dyer, University of Wisconsin-Madison

Overview

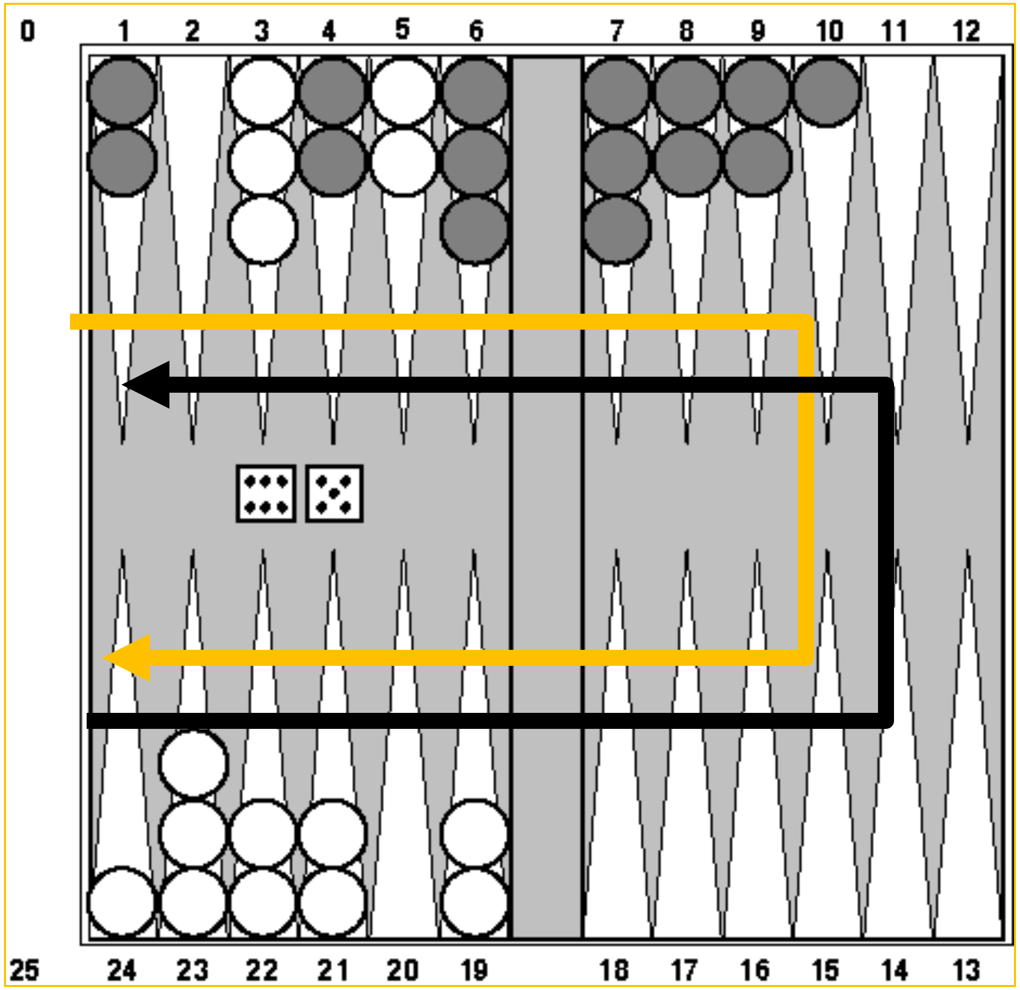
- Stochastic games
- Other issues
- AlphaGo Zero
- General game playing

Stochastic Games

- In real life, unpredictable external events can put us into unforeseen situations
- Many games introduce unpredictability through a random element, such as the throwing of dice
- These offer simple scenarios for problem solving with **adversaries and uncertainty**

Example: Backgammon

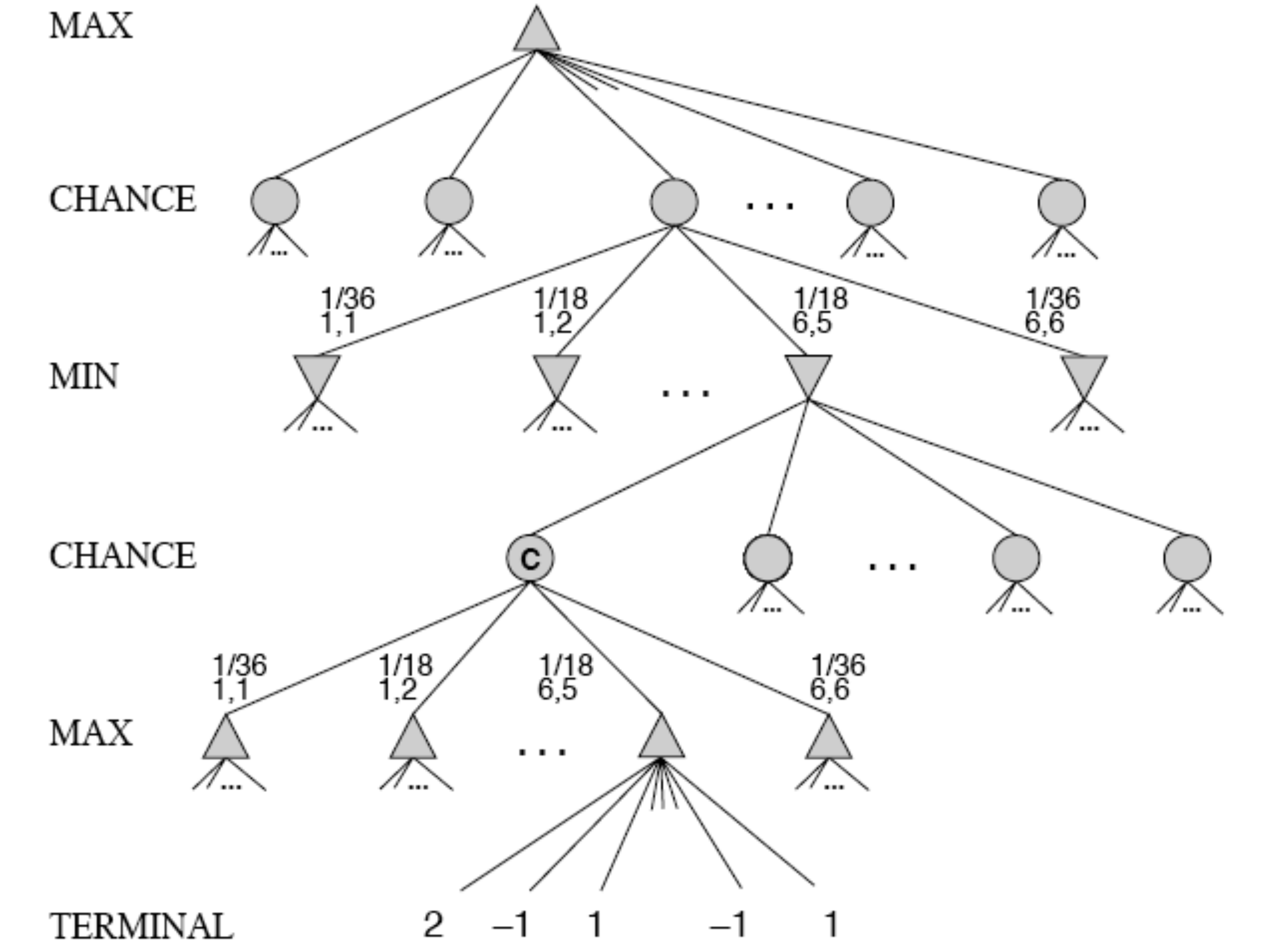
- Popular two-player game with uncertainty
- Players roll dice to determine what moves can be made
- White has just rolled 5 & 6, giving four legal moves:
  - 5-10, 5-11
  - 5-11, 19-24
  - 5-10, 10-16
  - 5-11, 11-16
- Good for exploring decision making in adversarial problems involving skill **and** luck



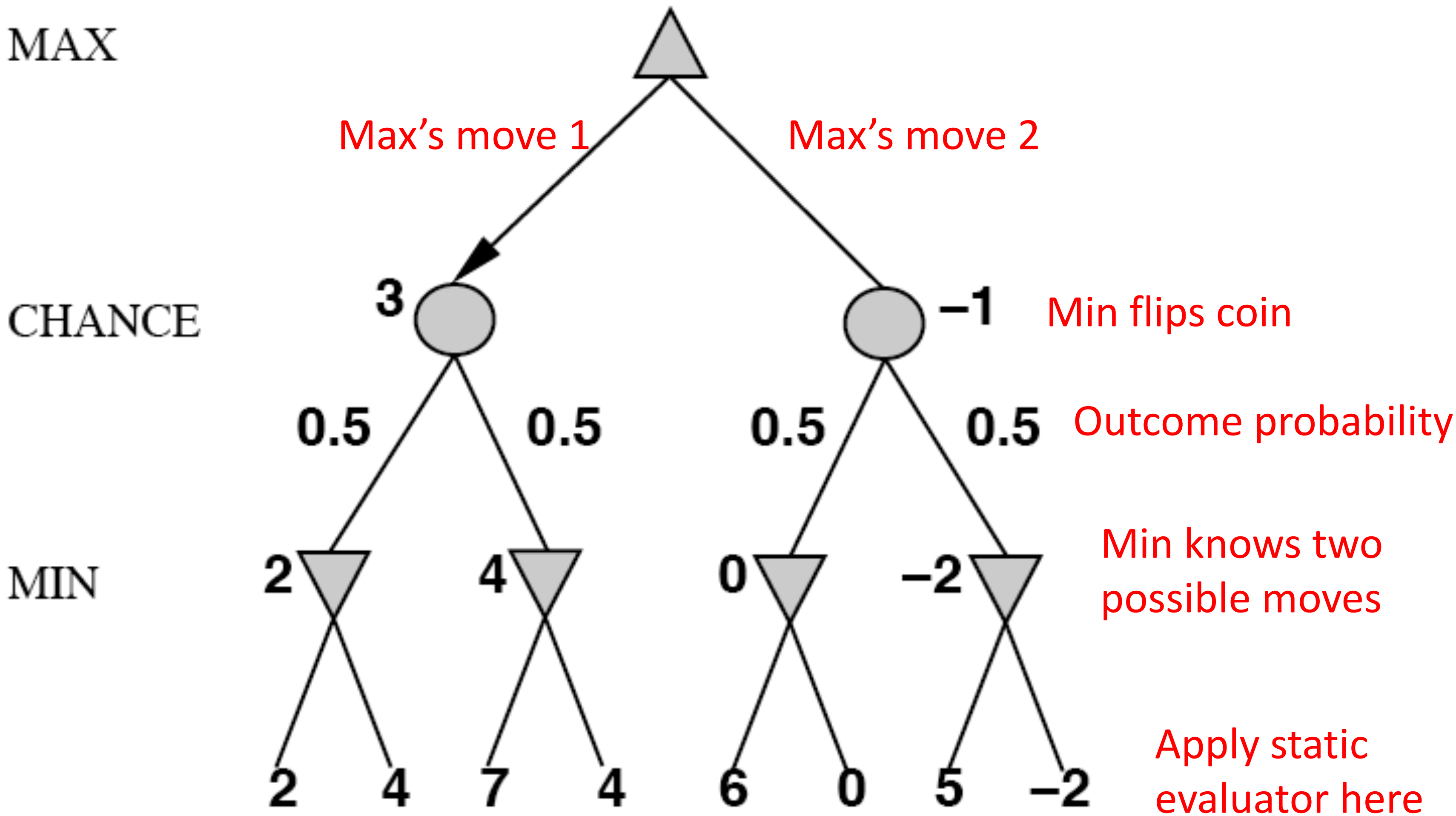
Why can’t we use MiniMax?

- Before player chooses a move, she rolls dice and **only then** knows what moves are possible
- Immediate outcome of each move also known
- But she doesn’t know the moves she or her opponent will have available in the **future**; they depend on the future dice rolls
- Need to adapt MiniMax to handle this
- Requires using probabilities & expected values

MiniMax trees with Chance Nodes



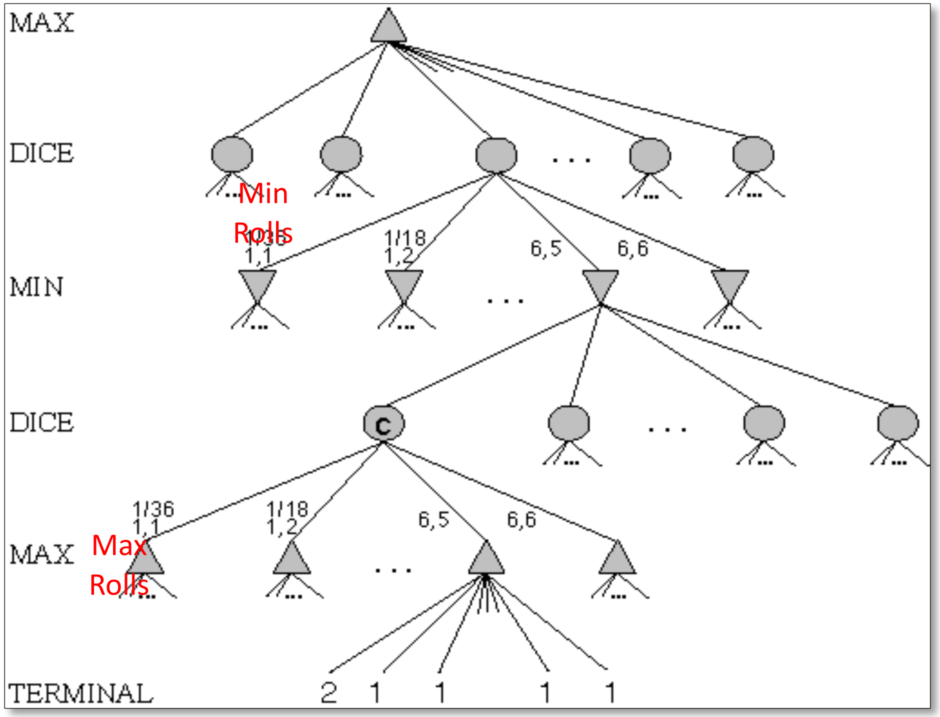
Understanding the notation



Board state includes chance outcome determining available moves

Game trees with chance nodes

- **Chance nodes** (circles) represent random events
- For random event with N outcomes, chance node has N children, each with a probability
- 2 dice: 21 distinct outcomes
- Use minimax to compute values for MAX and MIN nodes
- Use [expected values]() for chance nodes

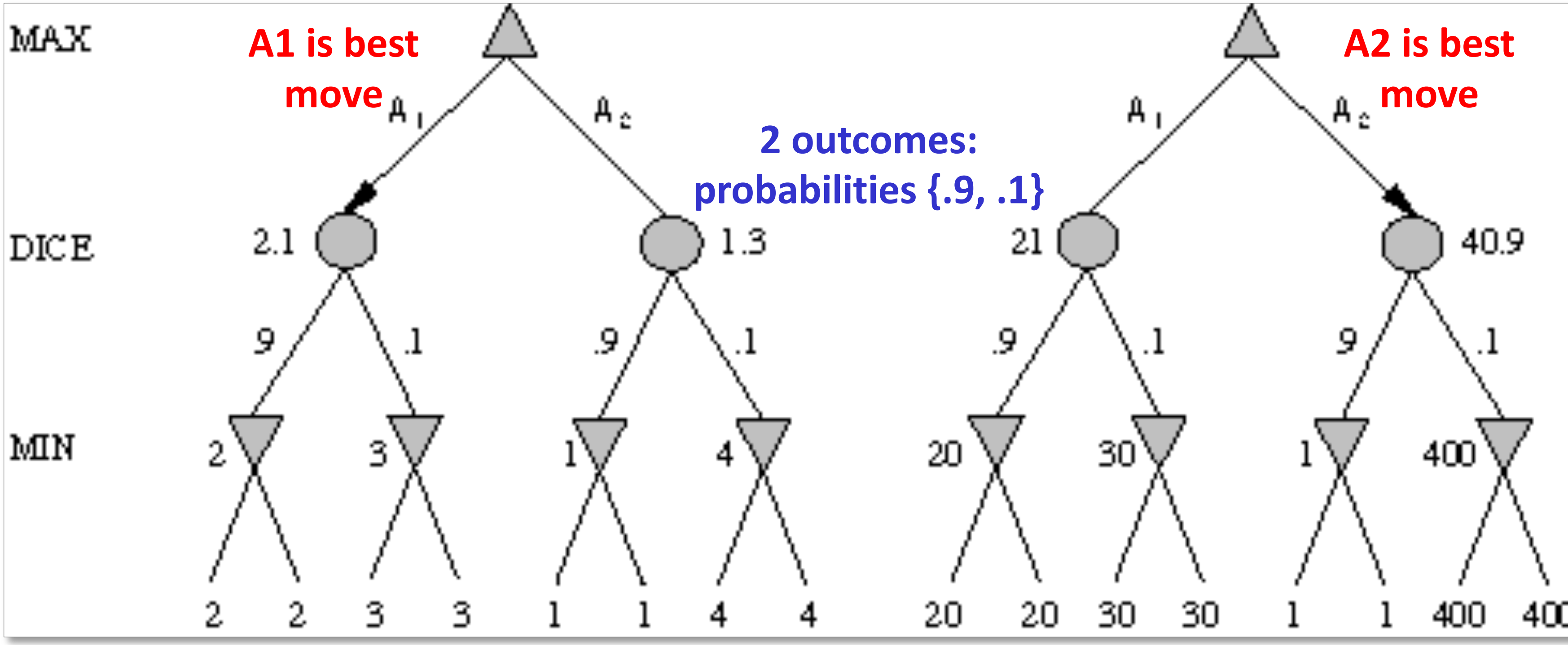


- Chance nodes over max node: expectimax(C) = $\sum_i(P(d_i)*\text{maxval}(i))$
- Chance nodes over min node: expectimin(C) = $\sum_i(P(d_i)*\text{minval}(i))$

Impact on lookahead

- Dice rolls **increase branching factor**
  - There are 21 possible rolls with two dice
- Backgammon: ~20 legal moves for given roll
  ~6K with 1-1 roll (get to roll again!)
- At depth 4: 20 * (21 * 20)**3 ≈ **1.2B boards !**
- As depth increases, probability of reaching a given node shrinks
  - lookahead's value diminished and alpha-beta pruning is much less effective
- TD-Gammon used depth-2 search + a good static evaluator to achieve world-champion level in the 1990s

Meaning of the evaluation function



- With probabilities & expected values we must be careful about **meaning of values** returned by static evaluator
- Bigger is better and twice as big is twice as good
- Relative-order preserving change of static evaluation values doesn't change minimax decision, but could here
- Linear transformations are OK

Games of imperfect information

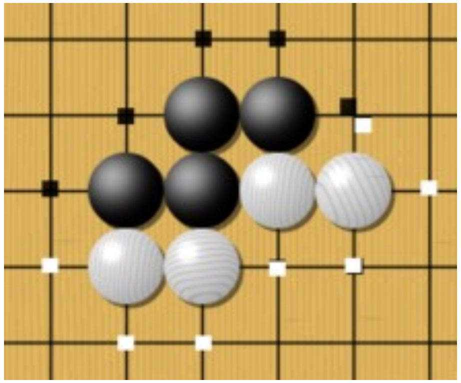
- E.g. card games where opponent's initial hand unknown
  - Can calculate probability for each possible deal
  - Like having one big dice roll at beginning of game
- Possible approach: minimax over each action in each deal; choose action with highest expected value over all deals
- Special case: if action optimal for all deals, it's optimal
- GIB bridge program from ~2000, approximates idea by
  1. Generating 100 deals consistent with bidding
  2. Picking action that wins most tricks on average

Other Issues

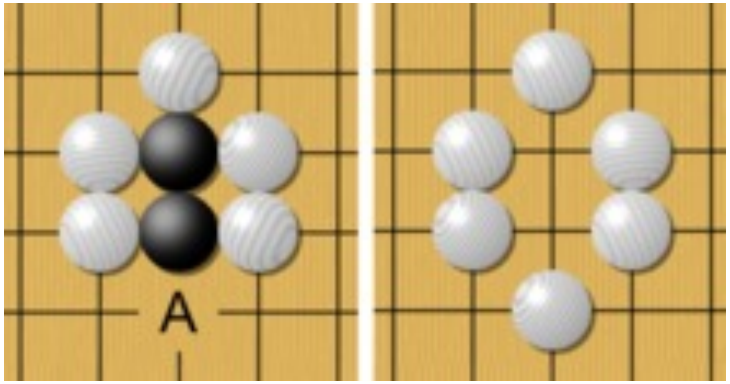
- Multi-player games, no alliances
  - E.g., many card games, like Hearts
- Multi-player games with alliances
  - Further complicated if alliances are fluid and can change during the game
  - E.g., Risk
  - More on this when we discuss game theory
  - Good model for a social animal like humans, where we must balance cooperation and competition

Go - the game

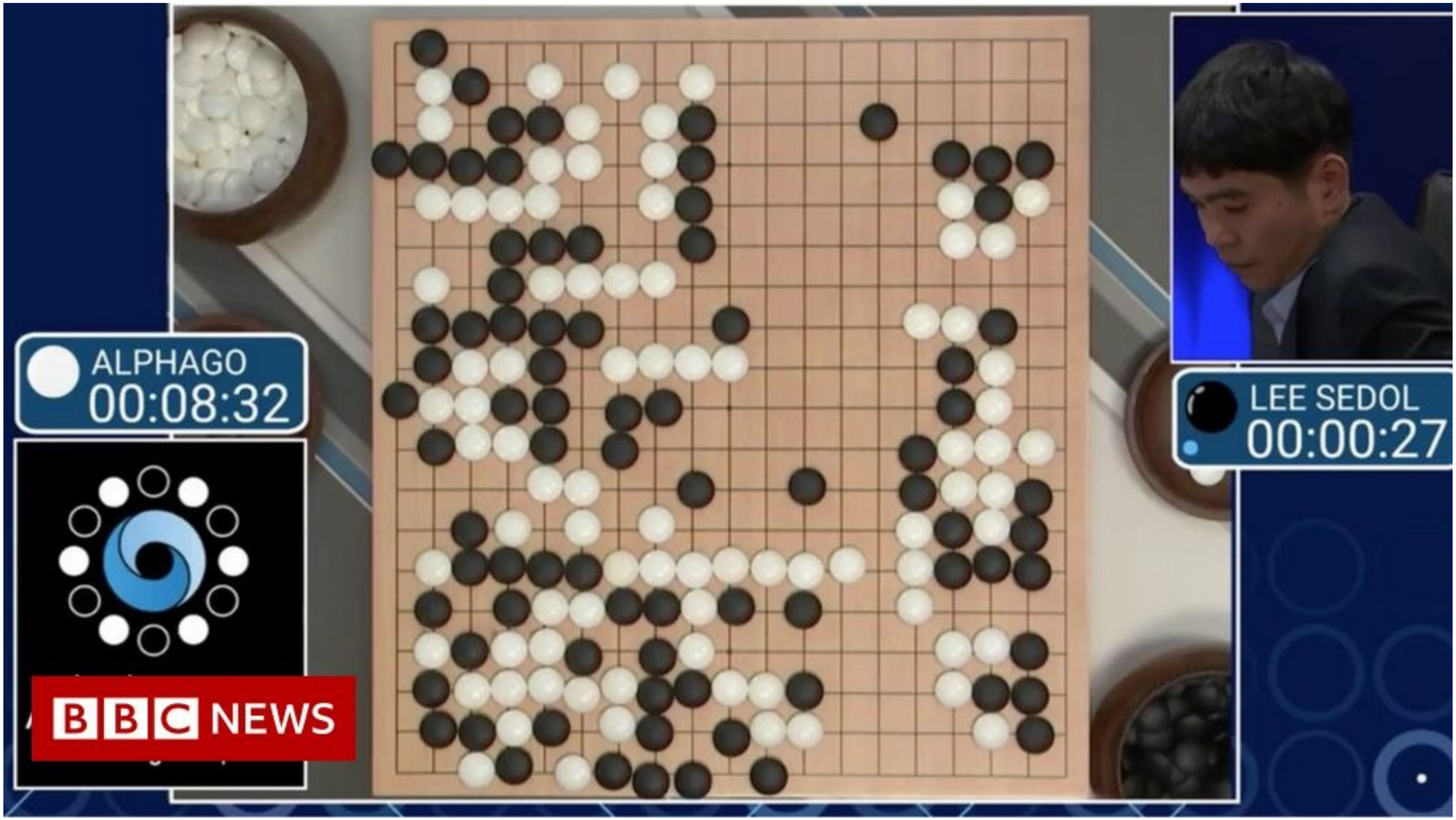
- Played on 19x19 board; black vs. white stones
- Huge state space $O(b^d)$: chess:$\sim 35^{80}$, go: $\sim 250^{150}$
- Rule: Stones on board must have an adjacent open point ("liberty") or be part of connected group with a liberty. Groups of stones losing their last liberty are removed from the board.



liberties



capture


ALPHAGO
00:08:32
LEE SEDOL
00:00:27
BBC NEWS

AlphaGO

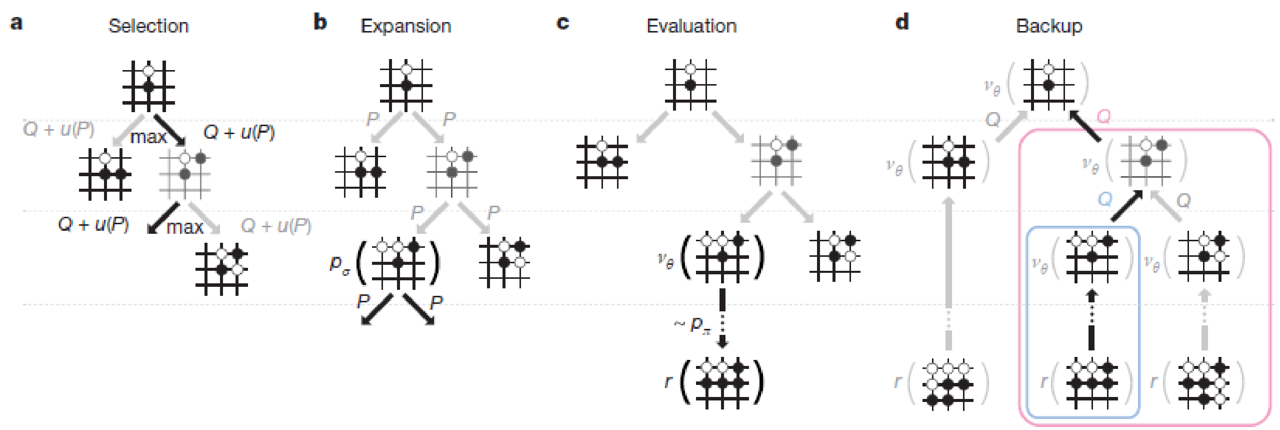
- Developed by Google's DeepMind
- Beat top-ranked human grandmasters in 2016
- Used Monte Carlo tree search over game tree

  expands tree via random sampling of search space
- *Science* Breakthrough of the year runner-up

  Mastering the game of Go with deep neural networks and tree search, Silver et al., *Nature*, 529:484–489, 2016
- Match with grandmaster Lee Sedol in 2016 was subject of award-winning 2017 AlphaGo

AlphaGo implementation

- Trained deep neural networks (13 layers) to learn **static evaluation function** and **policy function** to determine the best next move
- Performs Monte Carlo game search
  - explore state space like minimax
  - random "rollouts"
  - simulate probable plays by opponent according to policy function



AlphaGo implementation (~2016)

- Hardware: 1920 CPUs, 28O GPUs
- Neural networks trained in two phases over 4-6 weeks
- **Phase 1:** supervised learning from database of 30 million moves in games between two good human players
- **Phase 2:** play against versions of self using [reinforcement learning] to improve performance

Perspective on Games: Con and Pro

"Chess is the **Drosophila of artificial intelligence**. However, computer chess has developed much as genetics might have if the geneticists had concentrated their efforts starting in 1910 on **breeding racing Drosophila**. We would have some science, but mainly we would have very fast fruit flies."

John McCarthy, Stanford, 1990

"Saying Deep Blue doesn't really think about chess is like saying an airplane doesn't really fly because it doesn't flap its wings."

Drew McDermott, Yale, 1997

Why study games?

- Interesting, hard problems that require minimal "initial structure"
- Clear criteria for success
- A way to study problems involving {hostile, adversarial, competing} agents and the uncertainty of interacting with the natural world
- People have used them to assess intelligence
- Fun, good, easy to understand, PR potential
- Games often define very large search spaces
  - chess 35^{100} nodes in search tree, 10^{40} legal states