

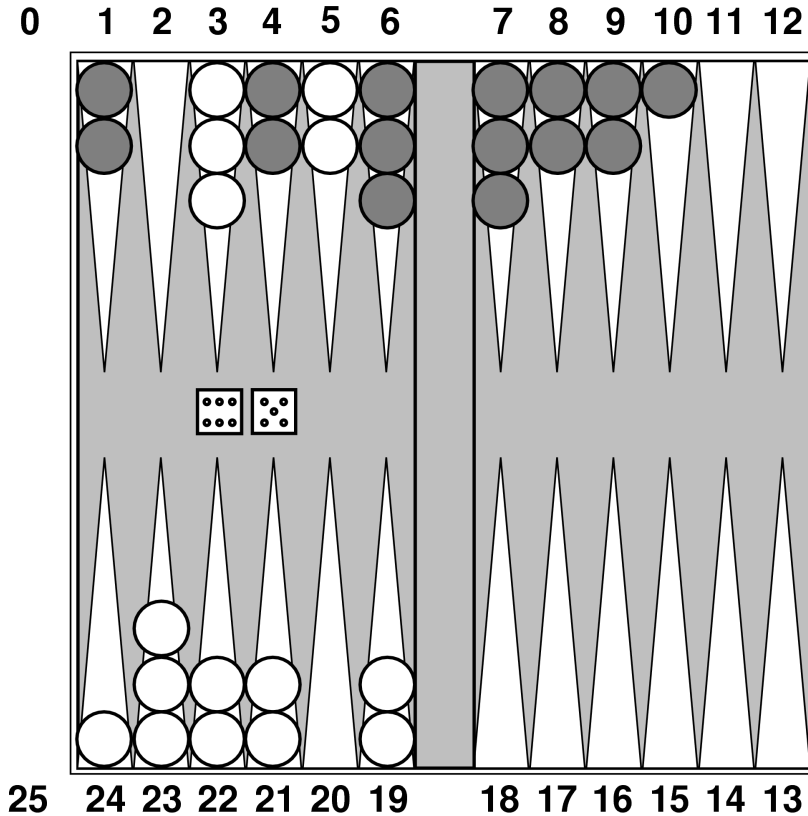
# Games That Involve Chance

- Games that include dice, playing cards, spinners, etc.
- When determining *the best move*, we don't know everything.
  - what roll of the dice will the opponent get.
  - what roll of the dice will we get on the next move.
  - what cards does the opponent have ?

# Use Probabilities

- In most situations we can compute the probability of
  - future events (dice rolls, getting specific cards, etc)
  - opponent state (what cards the opponent current has)
- We can modify minimax to deal with these situations.

# Backgammon



White has rolled 6,5.  
there are 4 possible moves:

5-10,5-11

5-11,19-24

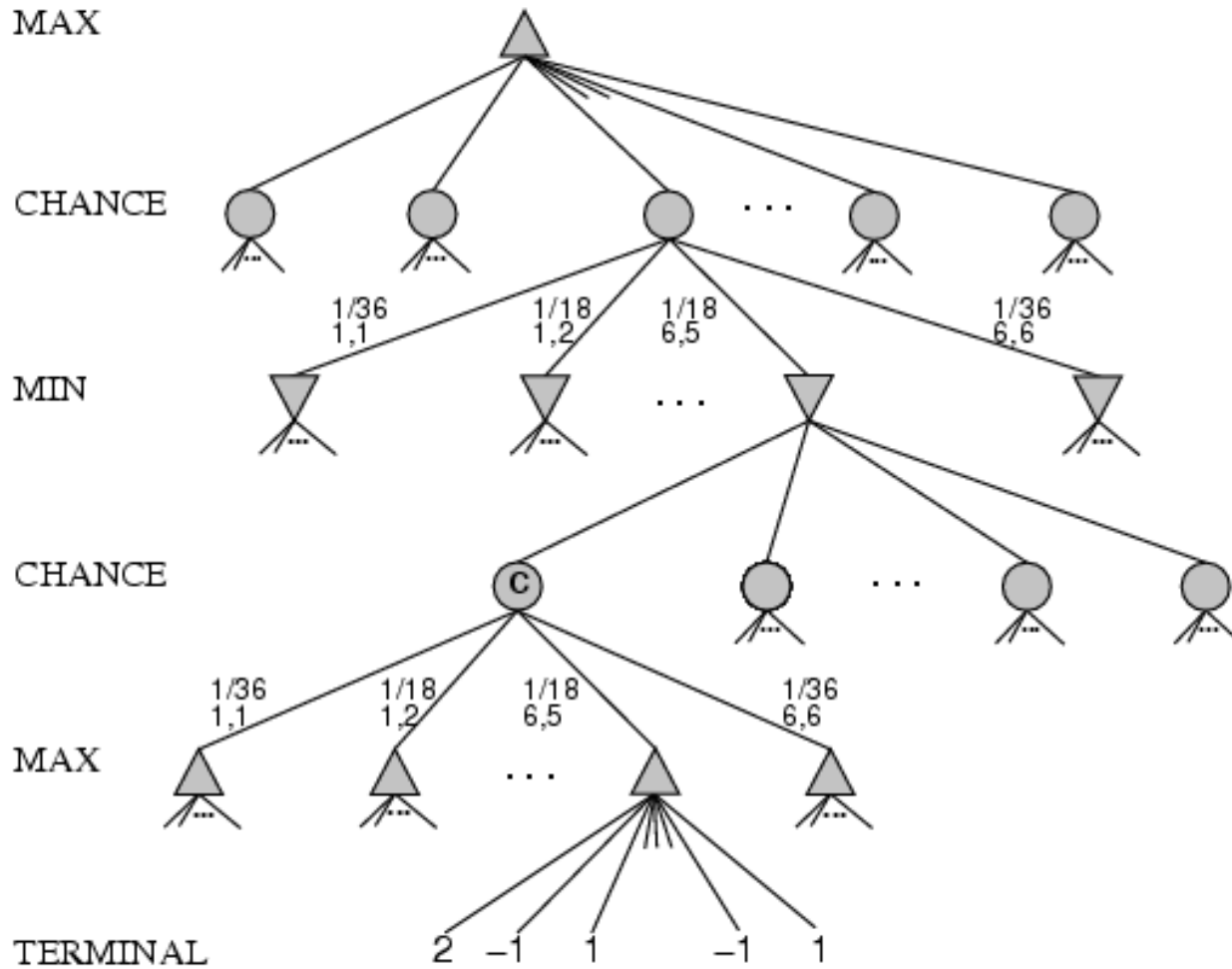
5-10,10-16

5-11,11-16

Which is best?

It depends on what the  
opponent rolls...

# Chance Nodes



# Chance Nodes

- Each child of a chance node is labeled with a probability
  - the sum of child probabilities must be 1.
- The general idea is to compute a score for each chance node based on the sum of the  $p(\text{child}) * \text{score}(\text{child})$ .
  - This is the *expected value*

# Expected-Minimax

EXPECTED-MINIMAX-VALUE( $n$ )=

UTILITY( $n$ )

If  $n$  is a terminal

$\max_{s \in \text{successors}(n)} \text{MINIMAX-VALUE}(s)$

If  $n$  is a max node

$\min_{s \in \text{successors}(n)} \text{MINIMAX-VALUE}(s)$

If  $n$  is a min node

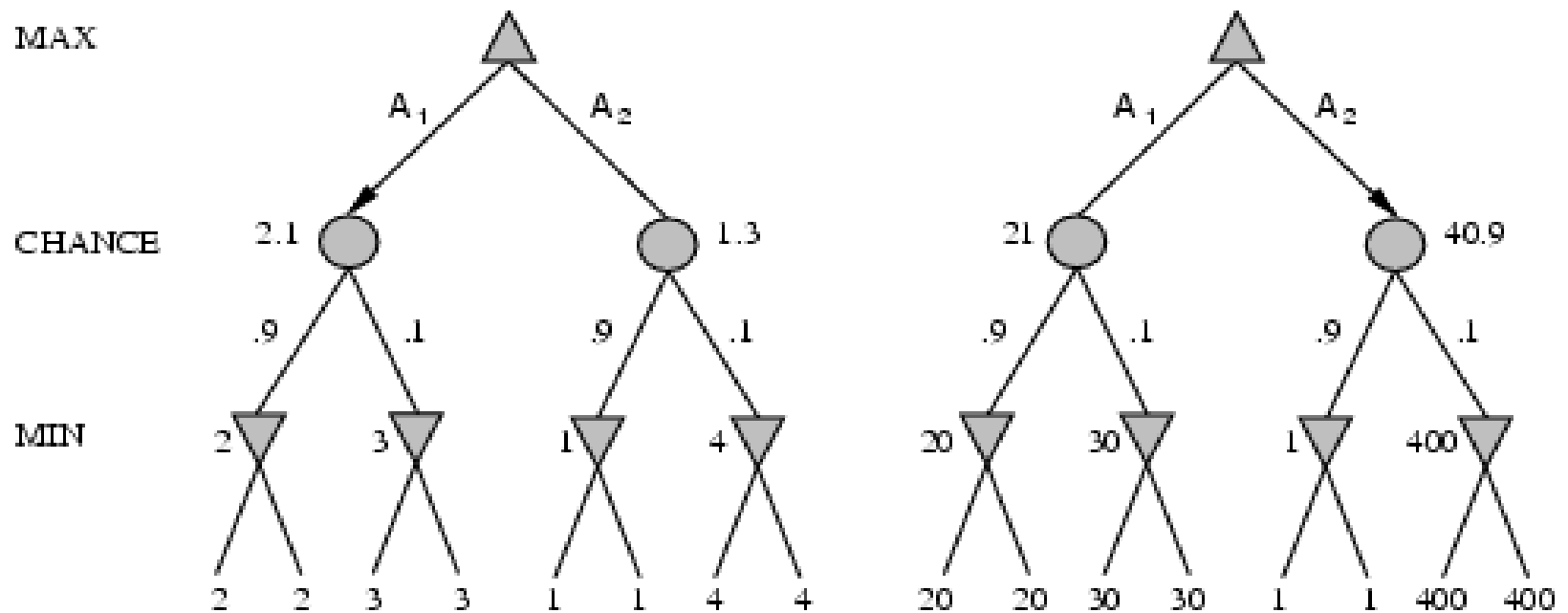
$\sum_{s \in \text{successors}(n)} P(s) \cdot \text{EXPECTEDMINIMAX}(s)$  If  $n$  is a chance node

# Evaluating Leaf Nodes

- Just like minimax
  - static evaluation function.
- Problem
  - relative order (ranking) of game states is no longer enough.
  - We need to worry about the actual values produced by the static evaluation function

# Potential Problem

- Identical *relative* ratings of game states.
- Two different *best moves*.



# Static Evaluation Function for use with Expected-Minimax

- To avoid the problem:
  - function must be a positive linear transformation of the probability of winning from a position.
    - this may be hard to develop!
    - recall that we want this function to be relatively simple...