Introduction to Game Theory

5. Lookahead Pathology

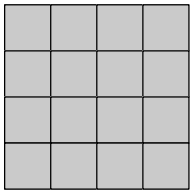
Dana Nau

University of Maryland

Motivation

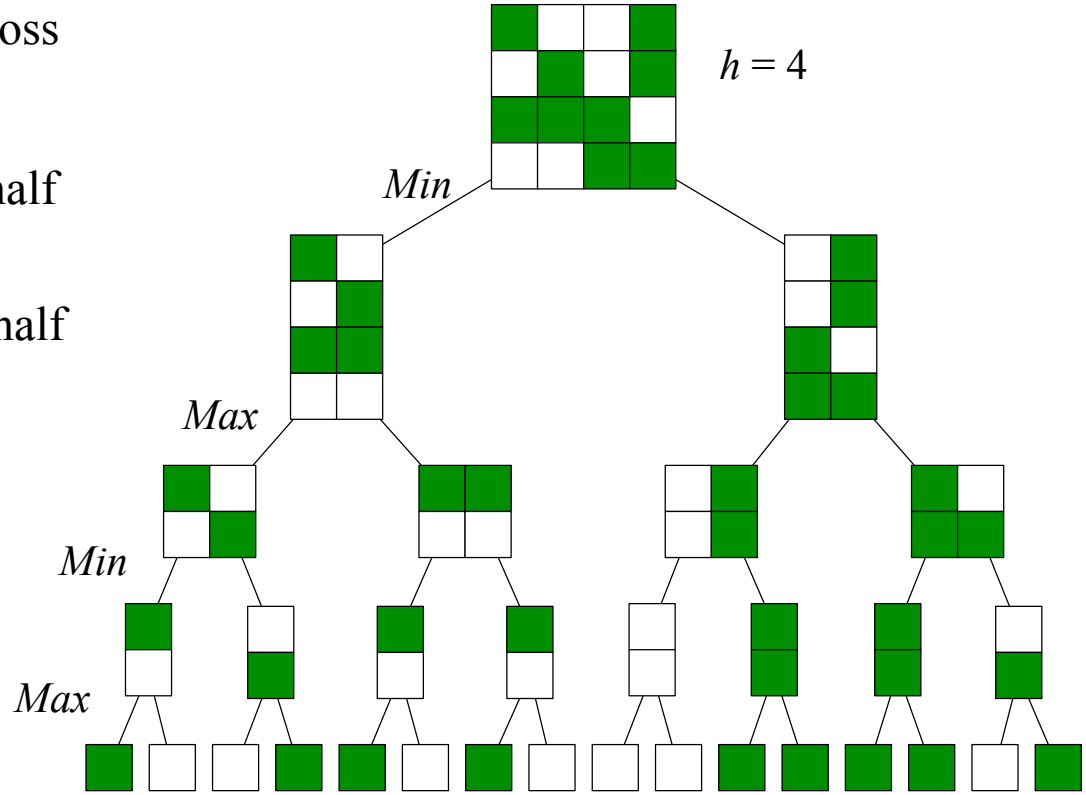
- When discussing game-tree search in the previous session, I said:
  - Deeper lookahead (i.e., larger depth bound *d*) usually gives better decisions
- For a many years, it was tacitly assumed that searching deeper would *always* give better decisions
  - For my Ph.D. work in 1979, I showed that's not true
  - There are infinitely many game trees for which searching deeper gives *worse* decisions

P-Games



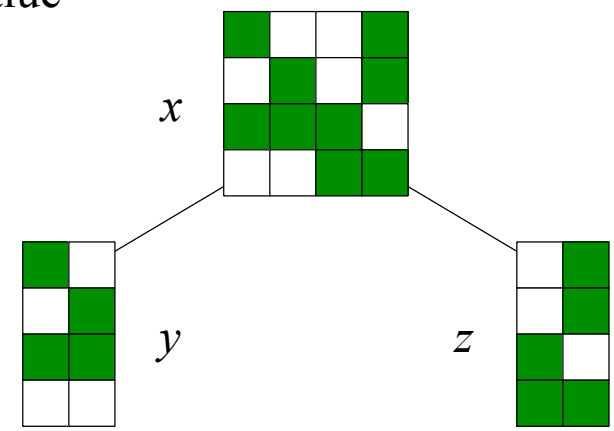
- A class of board-splitting games invented by Judea Pearl in 1980
- Playing board: chessboard of size $2^{\lfloor h/2 \rfloor} \times 2^{\lceil h/2 \rceil}$ instead of 8×8
  - (or equivalently, a string of 2^h squares)

- Initial state: randomly label each square as "win" or "loss"
  - I'll use green for win, white for loss
- Agents move in alternation
  - 1st move: remove either the left half or right half of the board
  - 2nd move: remove either the top half or bottom half of the board
- Continue until just one square is left
  - "win" square => win for the last player
  - "loss" square => loss for the last player
- This gives us a game tree of height h



Critical Nodes

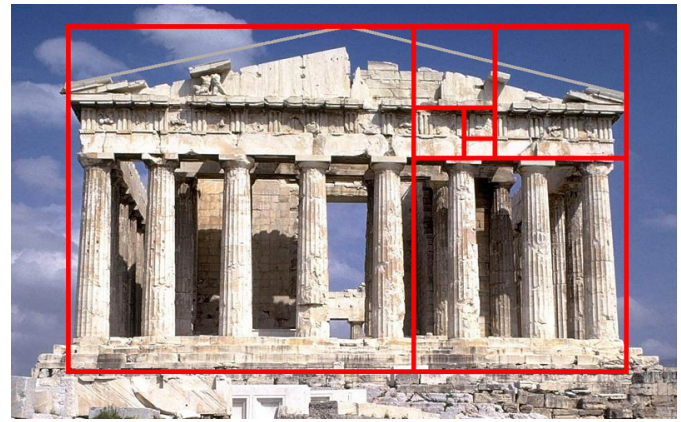
- Let x be a node in a P-game
  - Suppose x's **height** (number of moves from the end of the game) is h
- In order to talk about whether a deeper search at x gives a better or worse decision, x must be a node where the decision makes a difference
  - x's children shouldn't have the same minimax value
- x is **critical** if
  - it has a "loss" child y, i.e., $u^*(y) = -1$
  - and a "win" child z, i.e., $u^*(z) = 1$

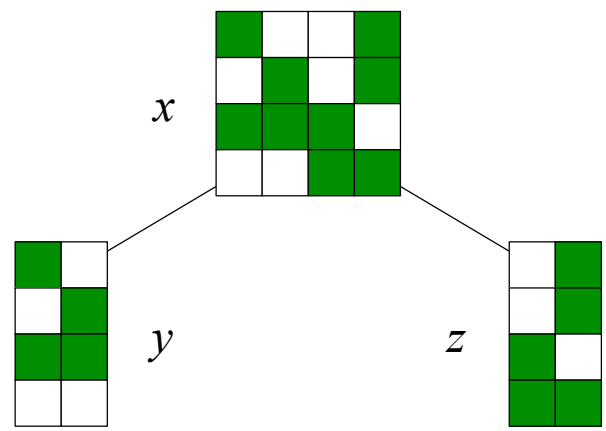


- Let $D(d,h) = P$(choose the "win" child | minimax search to depth d from a critical node x of height h)
- Then $D(d,h) = P[\text{MINIMAX}(y,d-1) < \text{MINIMAX}(z,d-1)] + 0.5\, P[\text{MINIMAX}(y,d-1) = \text{MINIMAX}(z,d-1)]$
  - where y and z are x's loss child and win child

Probability of a Win Node

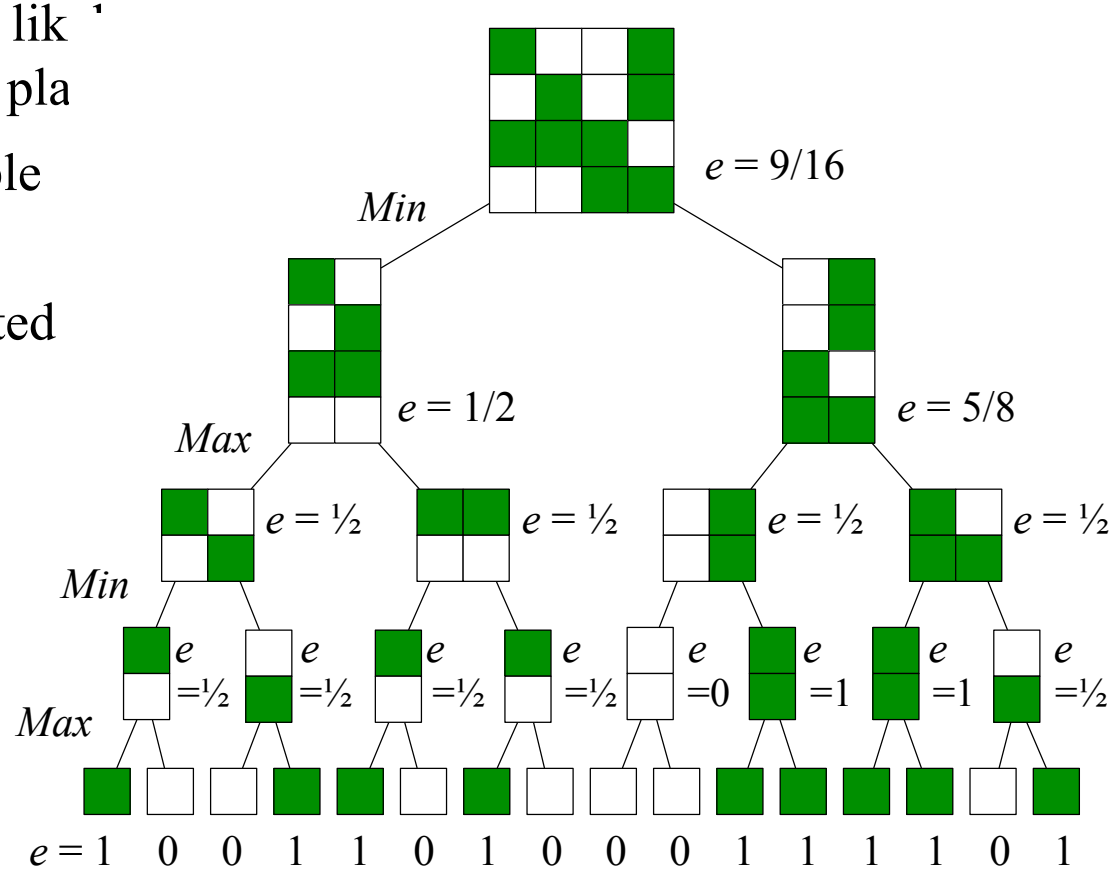
- Let $w = (3 - \sqrt{5})/2 \approx 0.382$
  - i.e., $w = 2 - \varphi = 1 - 1/\varphi$, where φ is the golden ratio
- Suppose we assign a "win" or "loss" label to each square at random, with probability p that a square is labeled "win"
- Let x be a node of height h, and y and z be its children
  - If $p > w$, then as we increase h, $P[y$ and z are both wins for the last player$] \rightarrow 1$
  - If $p < w$, then as we increase h, $P[y$ and z are both losses for the last player$] \rightarrow 1$
  - If $p = w$, then for all h, $P[u^*(y) \neq u^*(z)] = p(1-p)$
- So from now on, let $p = w$
  - This assures a reasonably good chance that a node at height h is critical





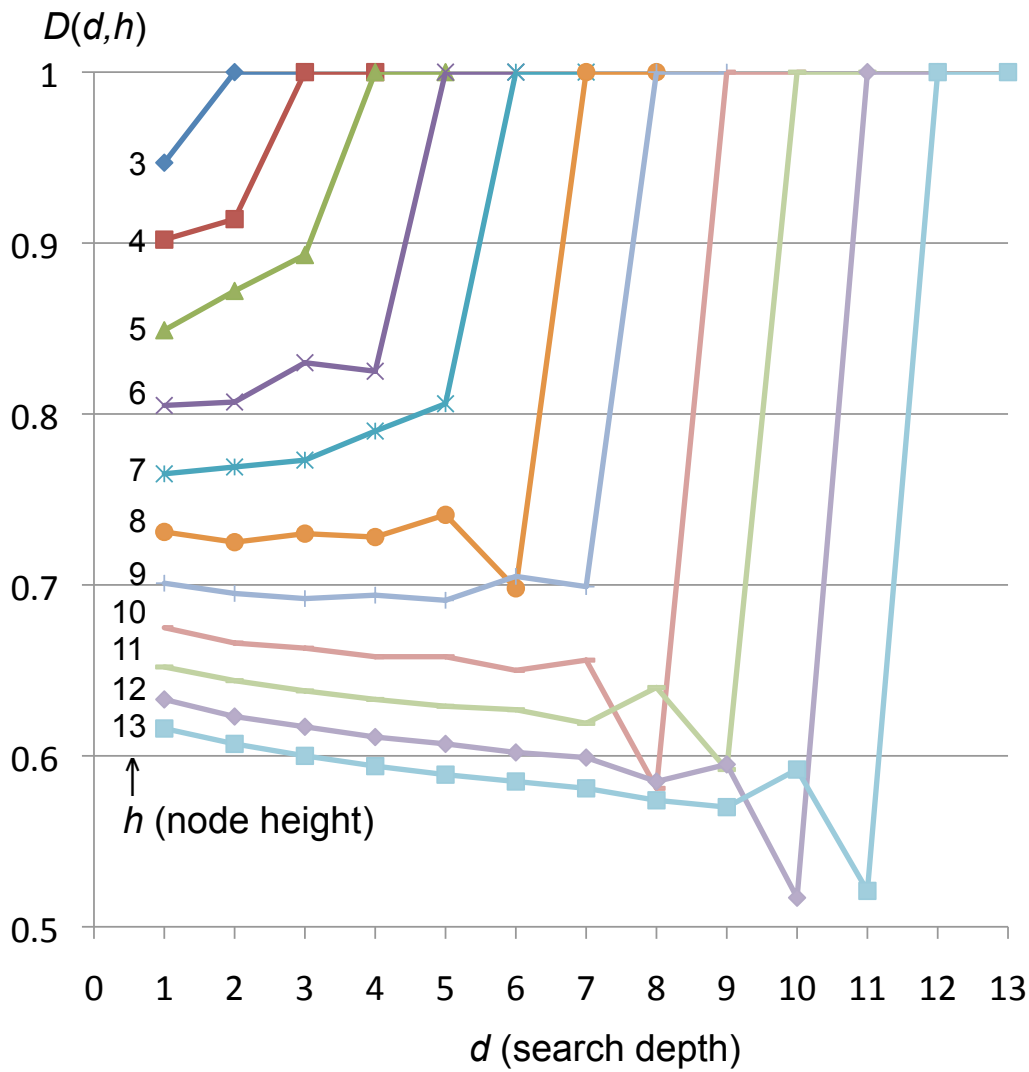
Evaluation Function

- Let $e(x)$ = (number of "win" squares) / (total number of squares)
  - The higher $e(x)$ is, the more likely that x is a win for the last player
  - The lower $e(x)$ is, the more lik that x is a win for the other pla
- Now that we have e, it's possible to derive a formula for $D(d,h)$
  - The derivation is complicated and I'll skip it
- But I'll show you the results



P-Games are Pathological

- If $d = h$, then $D(d,h) = 1$
  - i.e., searching to the game's end produces perfect play
- Likewise when $d = h–1$ (searching to just before the end)
- For node height $h \leq 7$, no pathology
  - $D(d,h)$ generally increases as we increase d
- For node height $h > 9$, there's lots of pathology
  - $D(d,h)$ generally decreases as we increase d

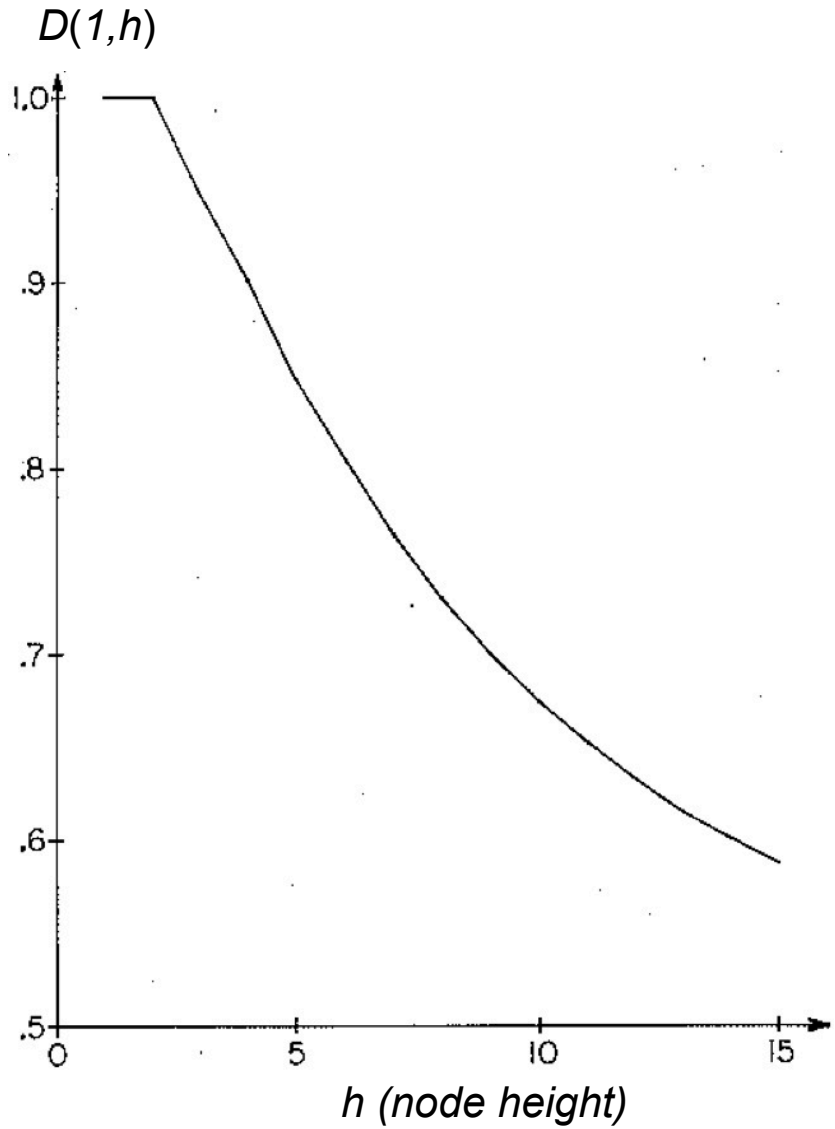


Why are the games pathological?

- **Hypothesis 1**: maybe it's due to the evaluation function
  - Let the **height** of a node be its distance from the end of the game
  - At a node of height h, a depth-d minimax search will apply the evaluation function e to nodes of height $h–d$
    - Increase the search depth d => decrease the node height $h–d$
    - If e is less accurate at nodes whose height is low, this could make $D(d,h)$ decrease as we increase d
  - To find out, let's measure e's accuracy as a function of node height
    - e's accuracy at a critical node x of height h
      $= P$[correct decision if we apply e directly to x's children]
      $= D(1,h)$
  - So let's look at $D(1,h)$ as $h \rightarrow 0$

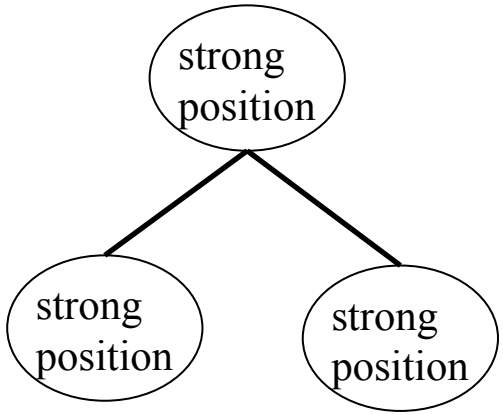
Why are the games pathological?

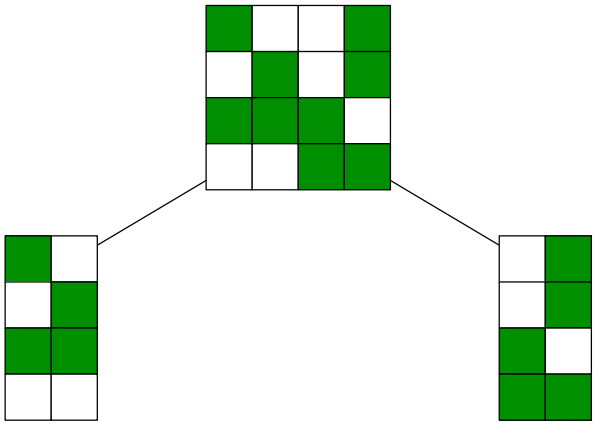
- The graph shows $D(1,h)$ as a function of h
- Notice that as $h \rightarrow 0$, $D(1,h) \rightarrow 1$
  - I.e., as x's height decreases, $e(x)$ gets ***more*** accurate
- Thus the hypothesis is wrong
  - The pathology isn't due to the evaluation function
  - It must be due to the game itself



Why are the games pathological?

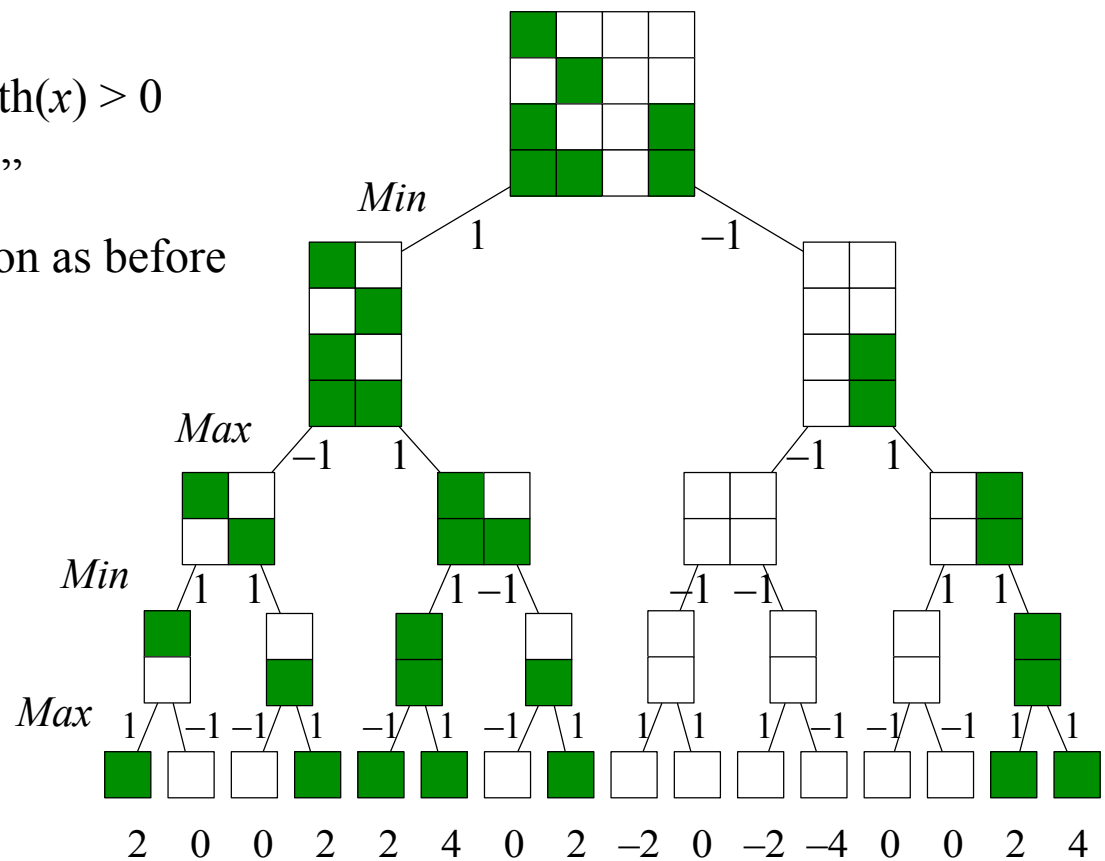
- **Hypothesis 2**:
- In most board games,
  - Some positions are "strong" (you're likely to win)
  - Others are "weak" (you're likely to lose)
  - Strong nodes are likely to have lots of strong children
    - So if a node is strong, that means its sibling nodes are probably strong too
  - Likewise for weak positions
- But in P-games, the values of sibling nodes are completely independent of each other
  - Could the pathology be due to that?
- Let's modify P-games to make sibling nodes have similar values





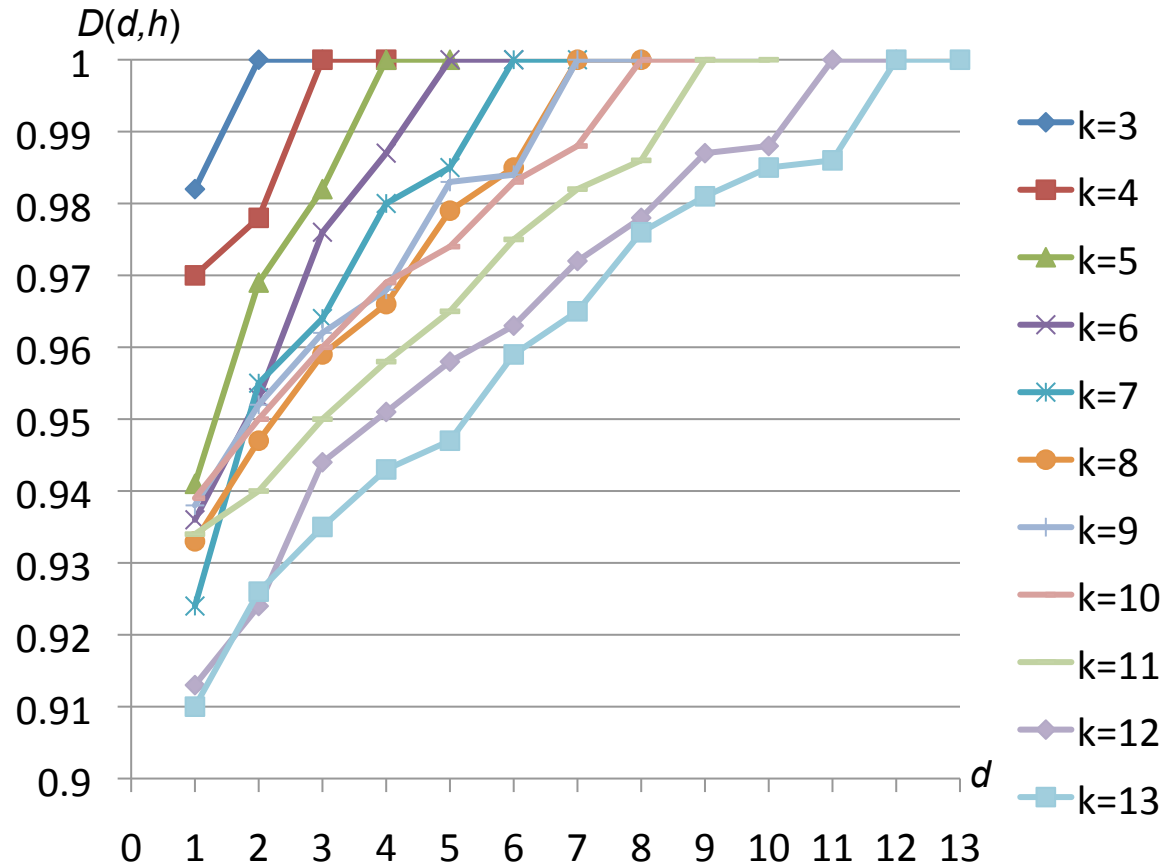
N-Games

- Everything is the same as in a P-game, except for how the board is initialized:
  - First assign 1 or –1 at random to each edge of the game tree
  - A node x's "strength" = sum of the edges on the path from the root to x
  - If x is a terminal node,
    - Label x "win" if strength(x) > 0
    - Otherwise label x "loss"
- Use the same evaluation function as before



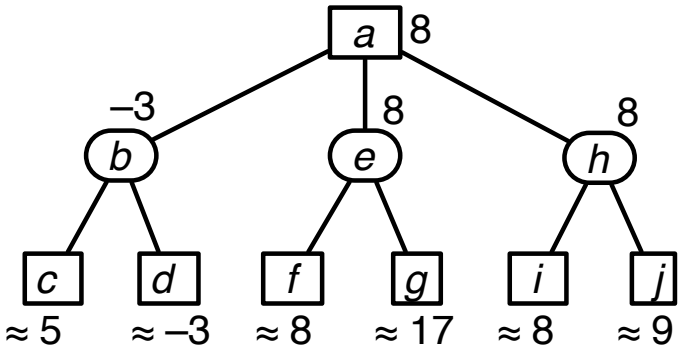
N-Games

- I don't know of a formula for computing $D(d,h)$ in N-games
  - So, Monte Carlo simulation instead
- For every combination of node height h and search depth d, I averaged $D(d,h)$ over 3200 randomly generated N-games
  - Result: at every node height h, searching deeper always helps
- So this suggests pathology is unlikely when there's a strong **local similarity** (correlation among sibling nodes)



Generalize to Other Games

- Suppose we do a minimax search to depth 2 at node *a*
  - *e* and *h* look equally good, and both look better than *b*
  - So we choose one of *e* and *h* at random, and move to it

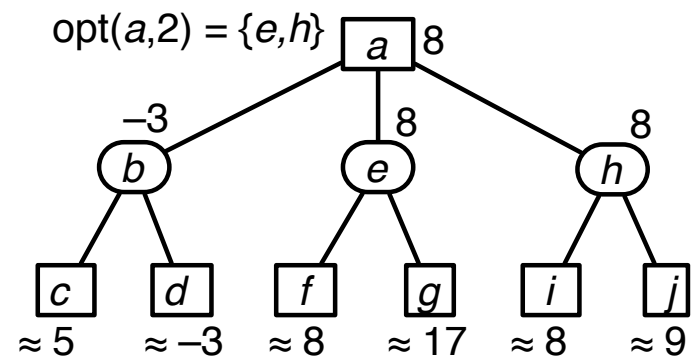


- What's the probability that we made a best move?

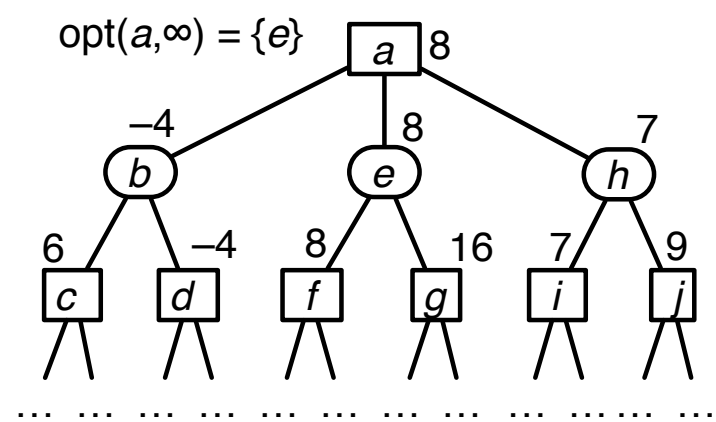
Probability of Optimal Decision

- For every node x, let $s(x)$ = {x's children}
- Let opt(x,d) = {the children of x that look best to a depth-d minimax search}

  = {y in $s(x)$ | minimax(x,d) = minimax(y,d–1)}
  - In the example, opt(a,2) = {e,h}



- The children of x that really are the best are the ones in opt(x,∞)
  - I.e., search to the end of the game
  - In the example, opt(a,∞) = {e}



- If we choose from opt(x,d) at random, then the probability of choosing an optimal move is
  - $P_{opt}(x,d) = |\text{opt}(x,d) \cap \text{opt}(x,\infty)| \,/\, |\text{opt}(x,d)|$
- In the example, $P_{opt}(a,2) = |\{e\}| \,/\, |\{e,h\}| = ½$

Degree of Pathology

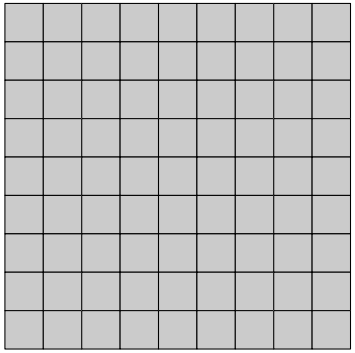
- The **decision error** at x is the probability that we didn't make the best choice:
  - $P_{err}(x,d) = 1 - P_{opt}(x,d)$

- The **degree of pathology** at x is the probability that searching deeper increases the decision error:
  - $p(x,i,j) = P_{err}(x,i) / P_{err}(x,j)$
  - where i and j are search depths, and $i > j$
- If $p(x,i,j) > 1$ then we have lookahead pathology at x
- A game G is considered **pathological** if $p(x,i,j)$, averaged over many x, is > 1
  - When G is pathological for some values of i and j, it usually is pathological for others

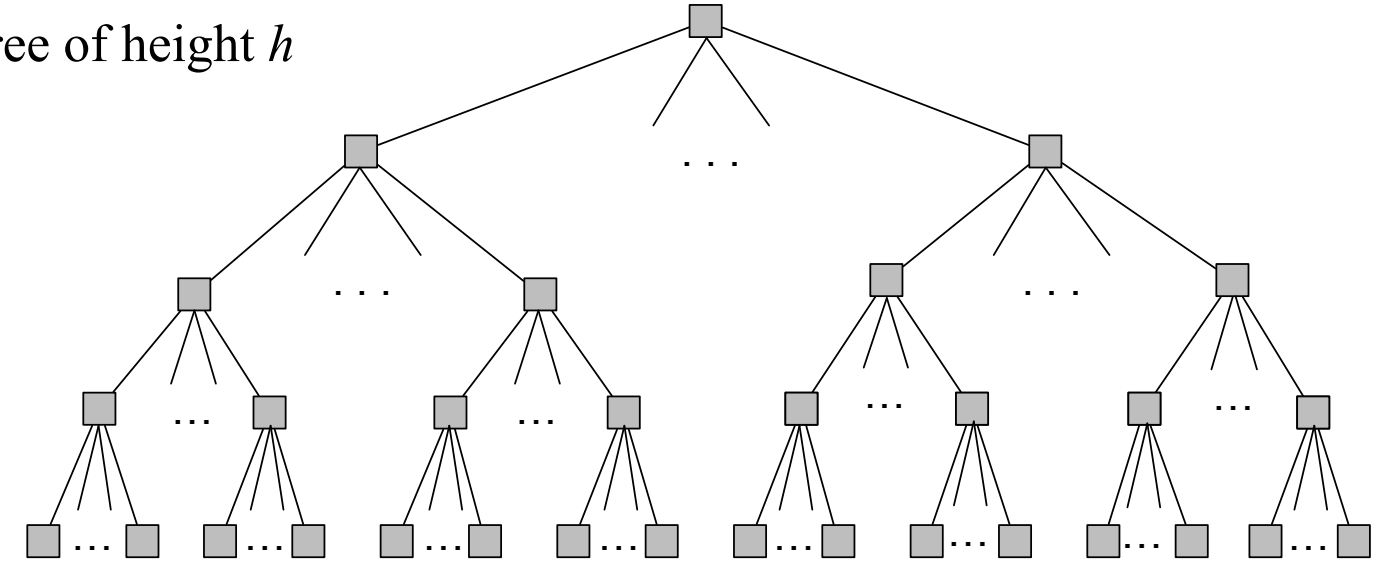
Influences on the Degree of Pathology

- Several factors affect the degree of pathology
- The most important ones:
  - Granularity
    - Number of possible utility values
  - Branching factor
    - Number of children of each node
  - Local similarity
    - Similarity among nodes that are close together in the tree
- There are several others
  - But most of them reduce to special cases of the ones above

How to Vary the Branching Factor



- Easy to get P-games and N-games of branching factor b
  - The board has size $b^{\lfloor h/2 \rfloor} \times b^{\lceil h/2 \rceil}$
    - (or equivalently, a string of b^h squares)
  - Each move: divide the board into b pieces instead of 2 pieces, and discard all but one of them
- Result: a b-ary tree of height h

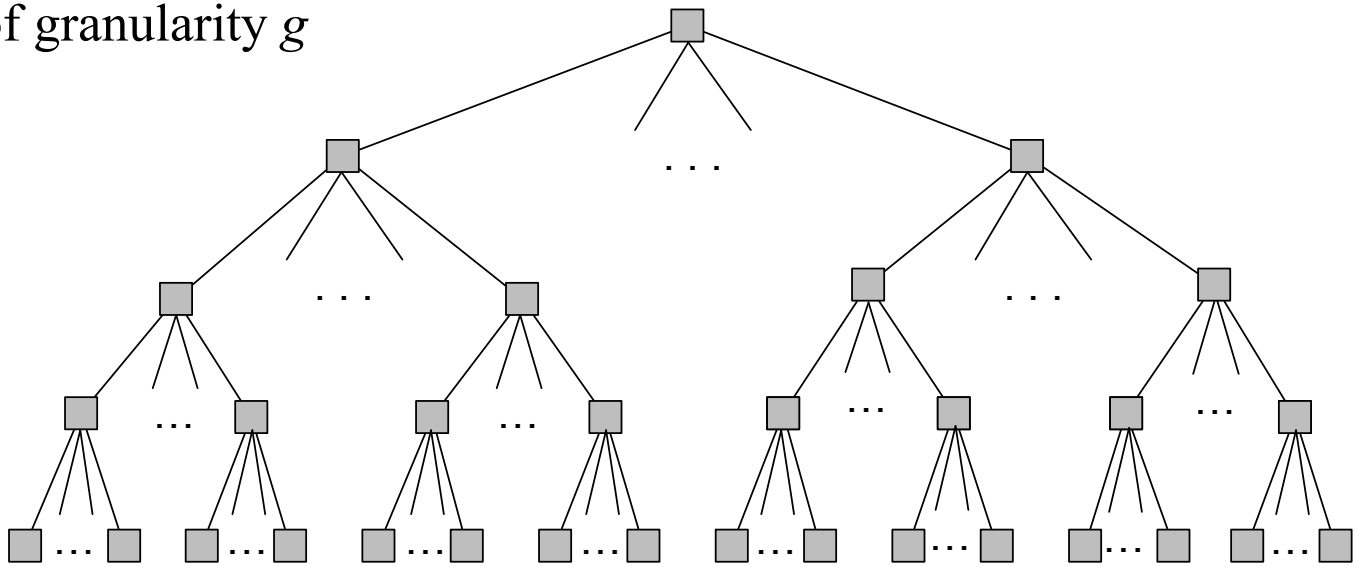


How to Vary the Granularity

- P-game with infinite granularity:
  - each square isn't "win" or "loss"
  - instead, its payoff is uniformly distributed over [0,1]
- N-game with infinite granularity:
  - Instead of assigning 1 or –1 to each edge, assign a random value from a normal (i.e., Gaussian) distribution
- P-game or N-game with granularity g:
  - Partition the interval [0,1] into g intervals of equal size

How to Vary the Local Similarity

- Use a parameter $0 \le s \le 1$ that determines the amount of local similarity:
- $s = 0$ => P-game of granularity g
- $s = 1$ => N-game of granularity g
- $0 < s < 1$ =>
  - Generate both P-game and N-game values for the nodes
  - For each terminal node, assign a payoff by making a random choice:
    - The node's P-game value with probability s, or its N-game value with probability $1-s$



Evaluation Function and Experiments

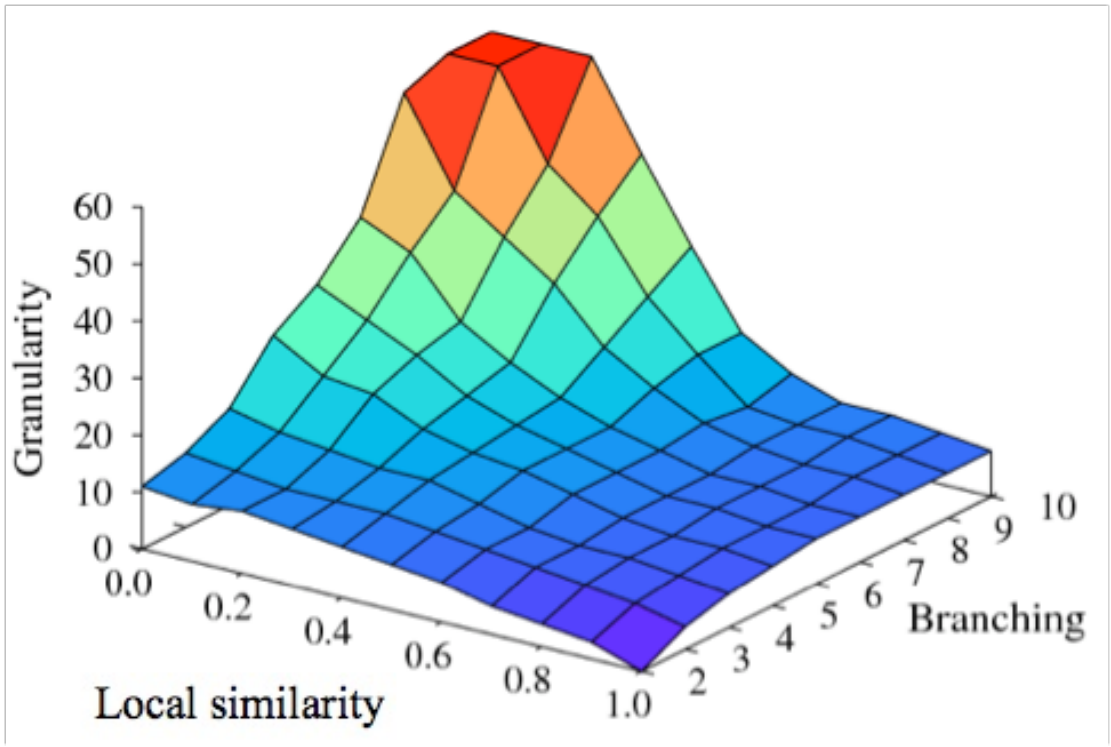
- So now we can vary b, g, and s independently
  - Experiments to measure how they influence the degree of pathology

    Nau, Luštrek, Parker, Bratko, and Gams.
    When Is It Better Not To Look Ahead?
    Artificial Intelligence, to appear.

- We can't use the previous evaluation function
  - It only works when $g = 2$
- Instead, use the following:
  - $e(x) = x$'s actual minimax value,
    corrupted by Gaussian noise with standard deviation $\sigma = 0.1$
  - For this evaluation function, accuracy is independent of node height

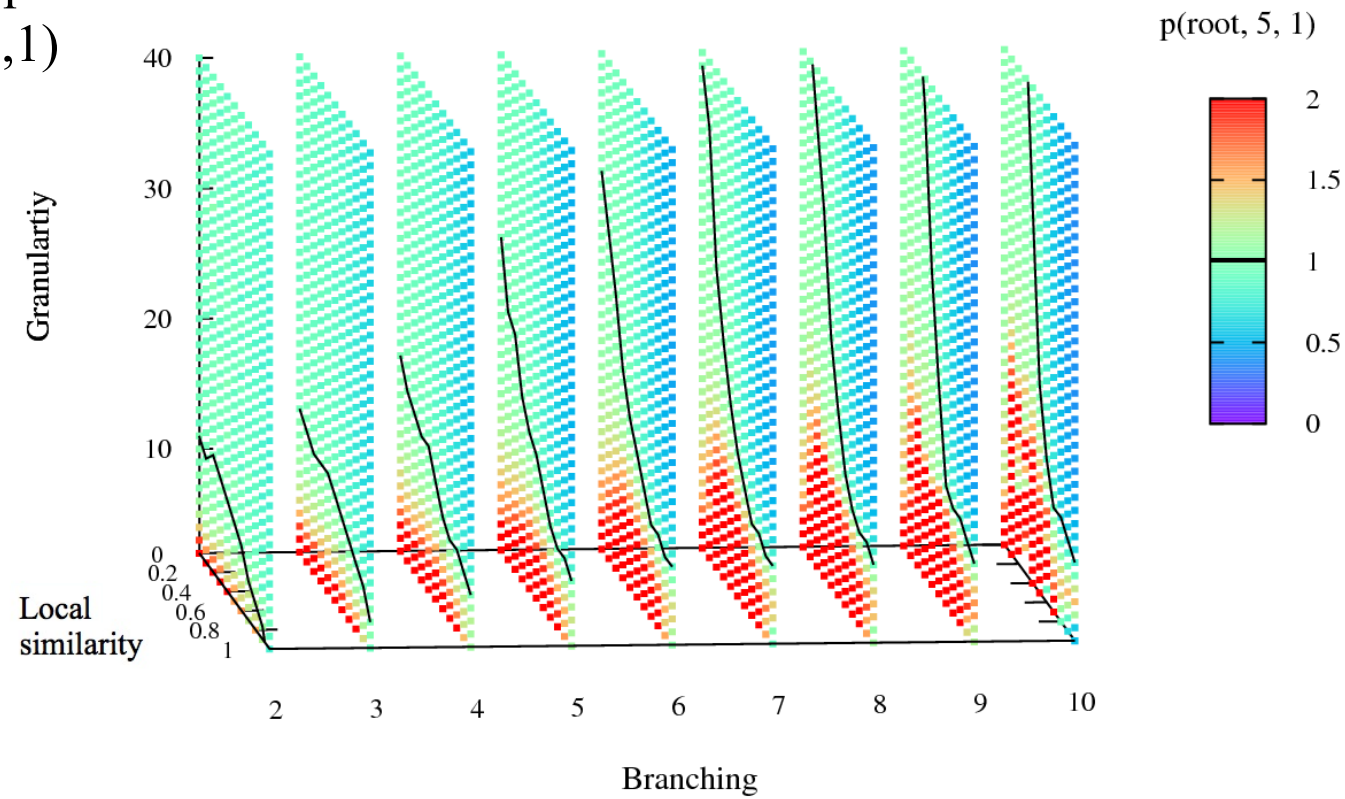
Granularity and Pathology

- Amount of granularity needed to avoid lookahead pathology
  - The space above the surface is pathological
  - The space below the surface is nonpathological



Branching Factor and Pathology

- The degree of pathology as a function of branching factor, granularity, and local similarity
  - Color of each point = value of $p(5,1)$
- Below the black lines: pathological
- Above the black lines: nonpathological.



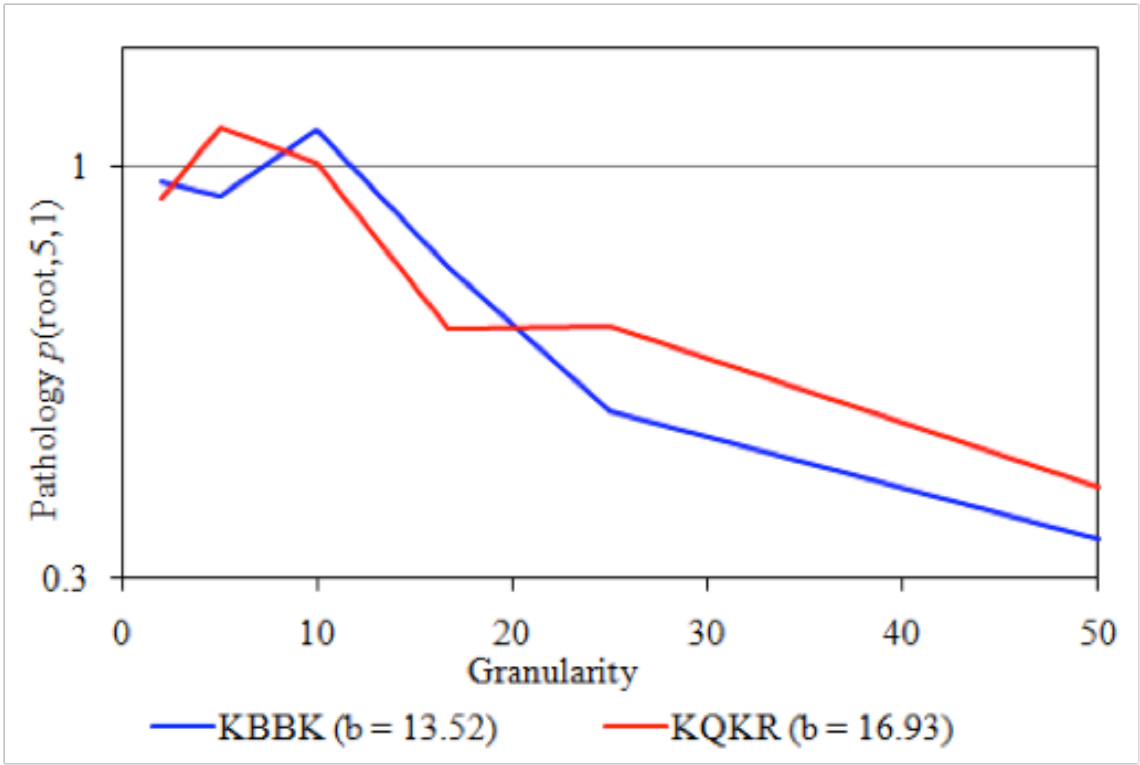
Does the Model Have Predictive Value?

- Does the model predict the trends in real games?
  - Yes!

- Let's look at
  - chess
  - kalah

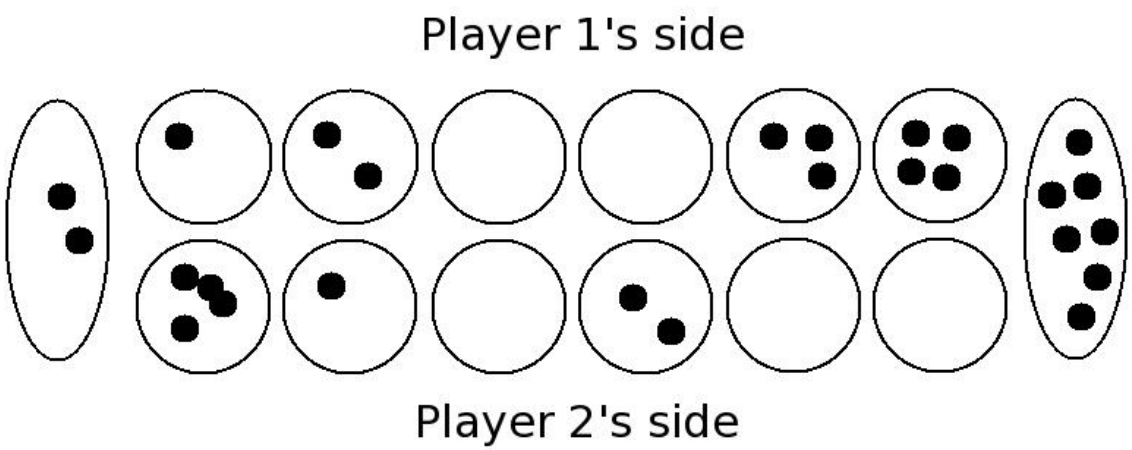
Chess endgames

- Degree of pathology as a function of granularity in
  - KBBK chess endgames (average b = 13.52 and cf = 0.58)
  - KQKR chess endgames (average b = 16.93 and cf = 0.37)



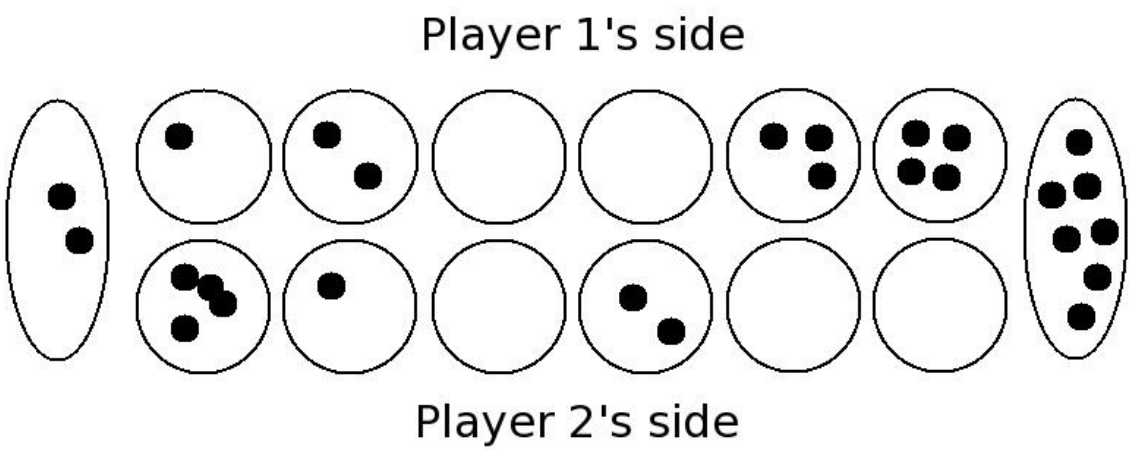
Kalah

- An ancient African game
- Moves:
  - Pick up the seeds in a pit on your side of the board
  - Distribute them, one at a time, to a string of adjacent pits
- Objective: acquire more seeds than the opponent, by either
  - moving them to your "kalah"
  - capturing them from the opponent's pits



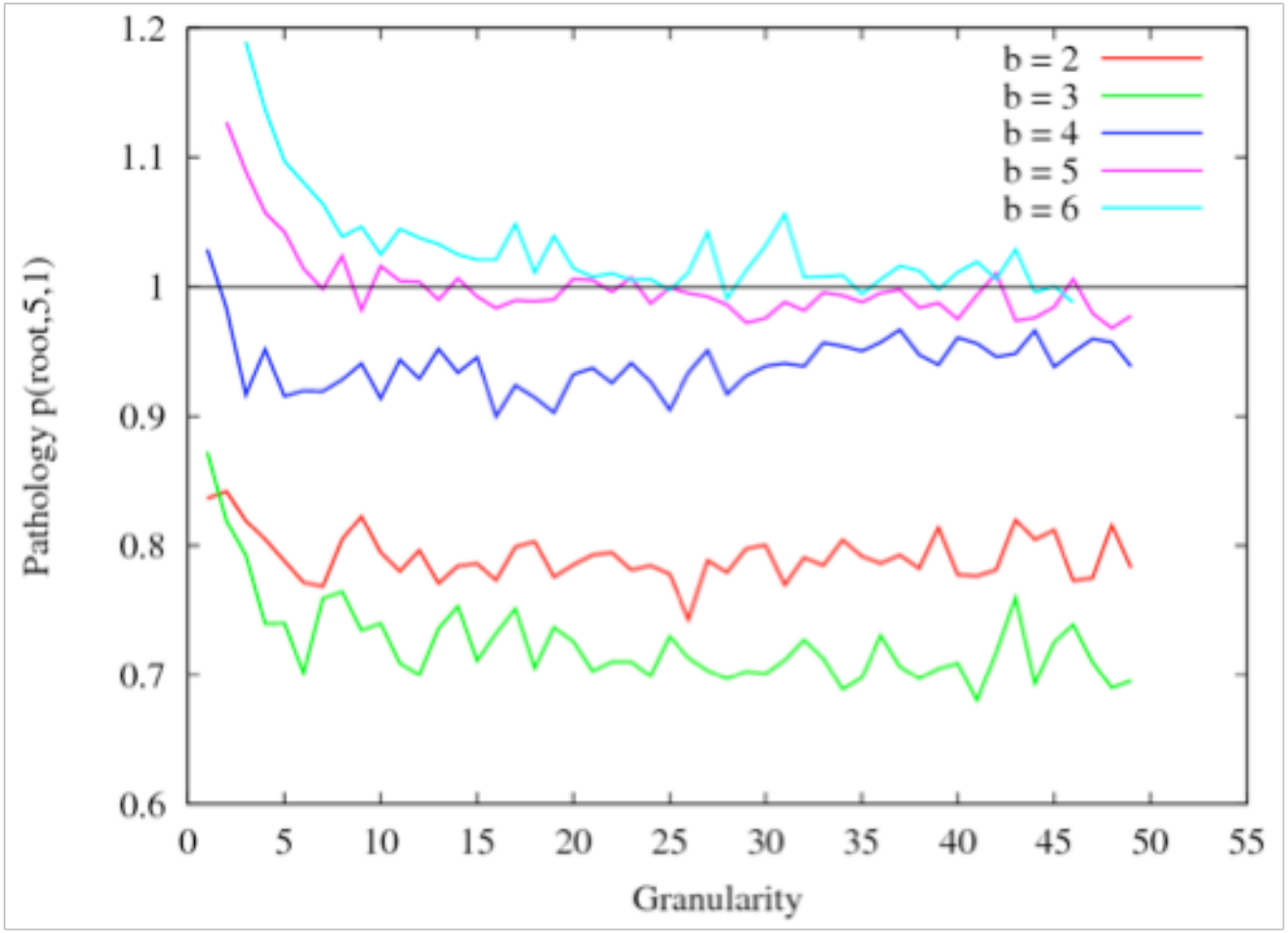
Modified Kalah

- Kalah is normally played until no seeds are left on the board
  - For computability, we limited the game to 8 moves
- To ensure a uniform branching factor
  - We allowed players to "move" from an empty pit
  - Such a move has no effect on the board
- We got different branching factors by varying the number of pits
- In Kalah, a player can move again if the last seed they placed lands in their kalah
  - We eliminated that rule, to get strict alternation of moves



Modified Kalah

- Degree of pathology in modified kalah as a function of granularity for several different branching factors

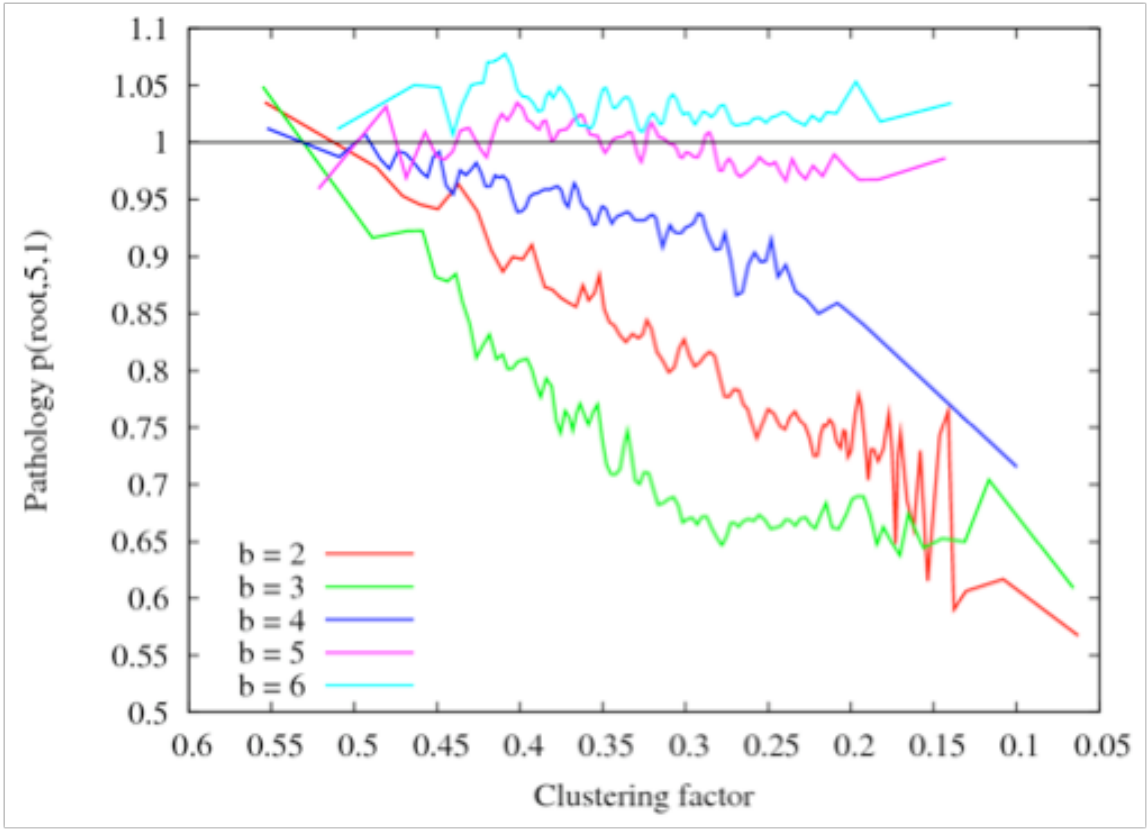


Modified Kalah

- The degree of pathology in modified kalah at several different branching factors, as a function of *clustering factor (cf)*

$$= \frac{\text{standard deviation of the sibling nodes' utilities}}{\text{standard deviation of the utilities throughout the game tree}}$$

  - Higher *cf* means less local similarity
  - Curves are smoothed for clarity



Summary

- In most game trees
  - Increasing the search depth usually improves the decision-making
- In pathological game trees
  - Increasing the search depth usually degrades the decision-making
- Pathology is more likely when
  - The branching factor is high
  - The number of possible payoffs is small
  - Local similarity is low
- Even in ordinary non-pathological game trees, *local* pathologies can occur
  - Work in progress: some of my students are developing algorithms to detect and overcome local pathologies