CS 4700:
Foundations of Artificial Intelligence

Bart Selman
selman@cs.cornell.edu

Module:
Adversarial Search
R&N: Chapter 5

Part II

Outline

Game Playing

Optimal decisions

Minimax

α-β pruning

Case study: Deep Blue

UCT and Go

Case Study: IBM's Deep Blue

Combinatorics of Chess

- Opening book
- Endgame
  - database of all 5 piece endgames exists; database of all 6 piece games being built
- Middle game
  - Positions evaluated (estimation)
    - 1 move by each player = 1,000
    - 2 moves by each player = 1,000,000
    - 3 moves by each player = 1,000,000,000

Positions with Smart Pruning

Search Depth (ply)		Positions
2		60
4		2,000
6		60,000
8		2,000,000
10	(<1 second DB)	60,000,000
12		2,000,000,000
14	(5 minutes DB)	**60,000,000,000**
16		2,000,000,000,000

How many lines of play does a grand master consider?

Around 5 to 7 ☺

Formal Complexity of Chess

How hard is chess?

- Obvious problem: standard complexity theory tells us nothing about finite games!
- **Generalized** chess to NxN board: optimal play is EXPTIME-complete
- Still, I would not rule out a medium-size (few hundred to a few thousand nodes) neural net playing almost perfect chess within one or two decades.

Game Tree Search (discussed before)

How to search a game tree was independently invented by Shannon (1950) and Turing (1951).

Technique called: MiniMax search.

Evaluation function combines material & position.

- **Pruning "bad" nodes: doesn't work in practice**
- **Extend "unstable" nodes (e.g. after captures): works well in practice.**

A Note on Minimax

Minimax "obviously" correct -- but

- **Nau (1982) discovered pathological game trees**

Games where

- **evaluation function grows more accurate as it nears the leaves**
- **but performance is worse the deeper you search!**

Clustering

Monte Carlo simulations showed clustering is important

- **if winning or loosing terminal leaves tend to be clustered, pathologies do not occur**
- **in chess: a position is “strong” or “weak”, rarely completely ambiguous!**

But still no completely satisfactory theoretical understanding of why minimax is good!

History of Search Innovations

Shannon, Turing	Minimax search	1950
Kotok/McCarthy	Alpha-beta pruning	1966
MacHack	Transposition tables	1967
Chess 3.0+	Iterative-deepening	1975
Belle	Special hardware	1978
Cray Blitz	Parallel search	1983
Hitech	Parallel evaluation	1985
Deep Blue	ALL OF THE ABOVE	1997

Evaluation Functions

Primary way knowledge of chess is encoded

- material
- position
  - doubled pawns
  - how constrained position is

Must execute quickly - constant time

- parallel evaluation: allows more complex functions
  - tactics: patterns to recognize weak positions
  - arbitrarily complicated domain knowledge

Learning better evaluation functions

- Deep Blue learns by tuning weights in its board evaluation function

$$f(p) = w_1 f_1(p) + w_2 f_2(p) + \dots + w_n f_n(p)$$

- Tune weights to find best least-squares fit with respect to moves actually chosen by grandmasters in 1000+ games. Weights tweaked multiple digits of precision.
- The key difference between 1996 and 1997 match!
- Note that Kasparov also trained on
  "computer chess" play. But, he did not have access to DB.

Transposition Tables

Introduced by Greenblat's Mac Hack (1966)

Basic idea: caching

- **once a board is evaluated, save in a hash table, avoid re-evaluating.**
- **called "transposition" tables, because different orderings (transpositions) of the same set of moves can lead to the same board.**

Transposition Tables as Learning

Is a form of root learning (memorization).

- **positions generalize sequences of moves**
- **learning on-the-fly**

Deep Blue --- huge transposition tables (100,000,000+), must be carefully managed.

Time vs Space

Iterative Deepening

- **a good idea in chess, as well as almost everywhere else!**
- **Chess 4.x, first to play at Master's level**
- **trades a little time for a huge reduction in space**
  - **lets you do breadth-first search with (more space efficient) depth-first search**
- **anytime: good for response-time critical applications**

Special-Purpose and Parallel Hardware

Belle (Thompson 1978)

Cray Blitz (1993)

Hitech (1985)

Deep Blue (1987-1996)

- Parallel evaluation: allows more complicated evaluation functions
- Hardest part: coordinating parallel search
- Interesting factoid: Deep Blue never quite played the same game, because of "noise" in its hardware!

Deep Blue

Hardware

- **32 general processors**
- **220 VSLI chess chips**

Overall: 200,000,000 positions per second

- **5 minutes = depth 14**

Selective extensions - search deeper at unstable positions

- **down to depth 25 !**

Aside:
4-ply ≈ human novice
8-ply to 10-ply ≈ typical PC, human master
14-ply ≈ Deep Blue, Kasparov (+ depth 25 for "selective extensions")

Evolution of Deep Blue

From 1987 to 1996

- faster chess processors
- port to IBM base machine from Sun
  - Deep Blue's non-Chess hardware is actually quite slow, in integer performance!
- bigger opening and endgame books
- 1996 differed little from 1997 - fixed bugs and tuned evaluation function!
  - After its loss in 1996, people underestimated its strength!

KASPAROV VERSUS DEEP BLUE

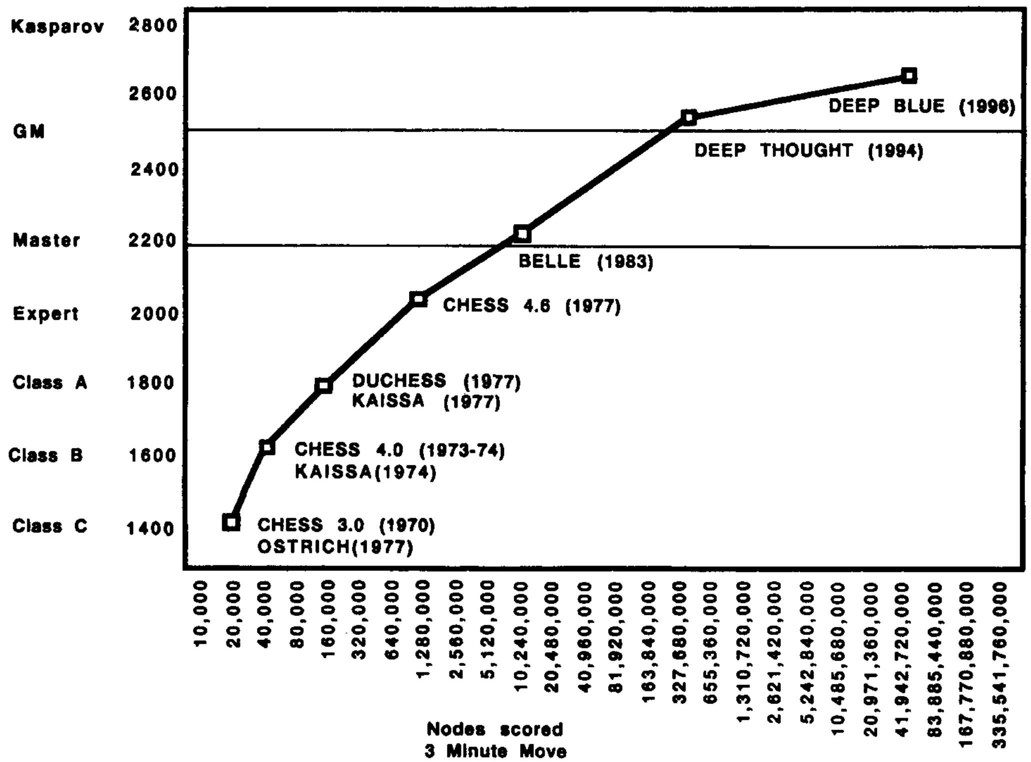


Figure 6.23. Relationship between the level of play by chess programs

Tactics into Strategy

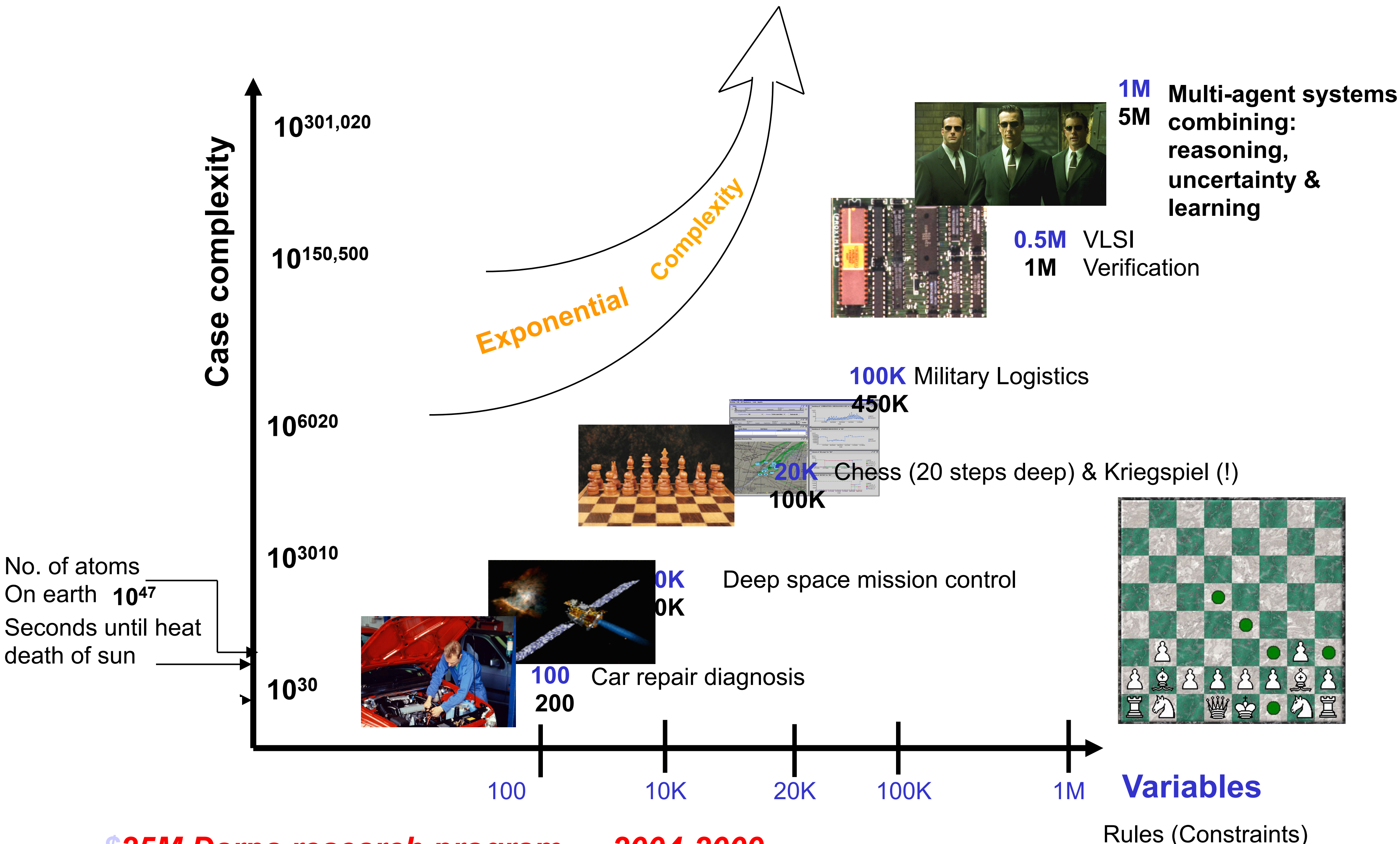
As Deep Blue goes deeper and deeper into a position, it displays elements of strategic understanding. Somewhere out there mere tactics translate into strategy. This is the closet thing I've ever seen to computer intelligence. It's a very weird form of intelligence, but you can feel it. It feels like thinking.

– Frederick Friedel (grandmaster), Newsday, May 9, 1997

This is an example of how massive computation ---
with clever search and evaluation function tuning ---
lead to a qualitative leap in performance (closer to human).

We see other recent examples with massive amounts of data and clever machine learning techniques. E.g. machine translation and speech/face recognition.

Automated reasoning --- the path



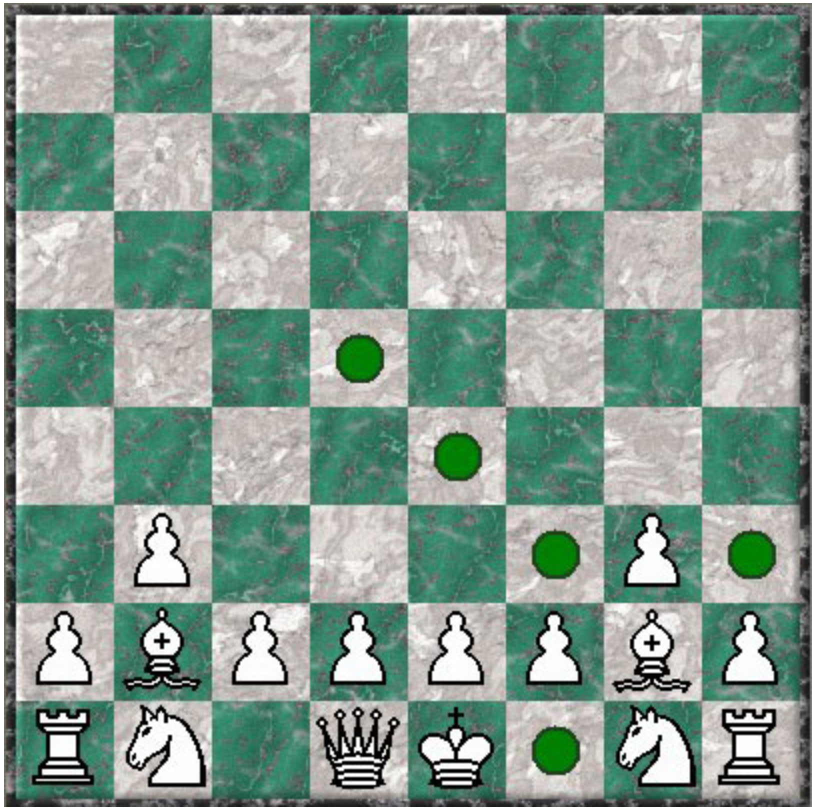
$25M Darpa research program --- 2004-2009

Kriegspiel

Pieces hidden from opponent

Interesting combination of reasoning, game tree search, and uncertainty.

Another chess variant: Multiplayer asynchronous chess.



The Danger of Introspection

When people express the opinion that human grandmasters do not examine 200,000,000 move sequences per second, I ask them, ``How do you know?'' The answer is usually that human grandmasters are not aware of searching this number of positions, or are aware of searching many fewer. But almost everything that goes on in our minds we are unaware of.

– Drew McDermott

In fact, recent neuroscience evidence shows that true expert performance (mind and sports) gets "compiled" to the sub-conscience level of our brain, and becomes therefore inaccessible to reflection. (Requires approx. 10K hours of practice for world-level performance.)

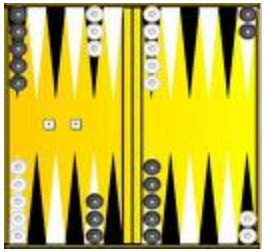
State-of-the-art of other games

Deterministic games in practice

Checkers: Chinook ended 40-year-reign of human world champion Marion Tinsley in 1994. Used a pre-computed endgame database defining perfect play for all positions involving 8 or fewer pieces on the board, a total of 444 billion positions.

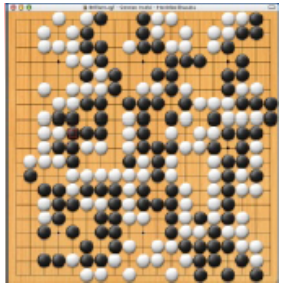
2007: proved to be a draw! Schaeffer et al. solved checkers for "White Doctor" opening (draw) (about 50 other openings).

Othello: human champions refuse to compete against computers, who are too strong.



Backgammon: TD-Gammon is competitive with World Champion (ranked among the top 3 players in the world). Tesauro's approach (1992) used learning to come up with a good evaluation function. Exciting application of reinforcement learning.

Playing GO



Go: human champio… …ompete against computers, consid… …In GO, b > 300, so most program… …knowledge bases to suggest pla… …&N, 2nd edition).

Not true!

Computer Beats Pro at U.S. Go Congress

http://www.usgo.org/index.php?%23_id=4602

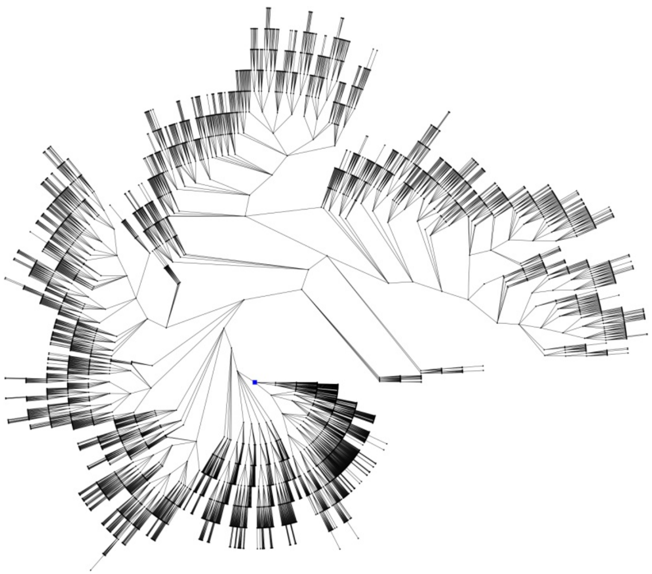
On August 7, 2008, the computer program MoGo running on 2… (se…) beat professional Go player Myungwan Kim (8p) in a hand… 19x19 board. The handicap given to the computer…

MoGo uses Monte Carlo based methods combi… *confidence bounds applied to t…*

"UCT may prove useful for targeting advertisements on the Web, finding the best settings for an industrial plant or optimizing channel allocation in cellular systems." – Scientific American

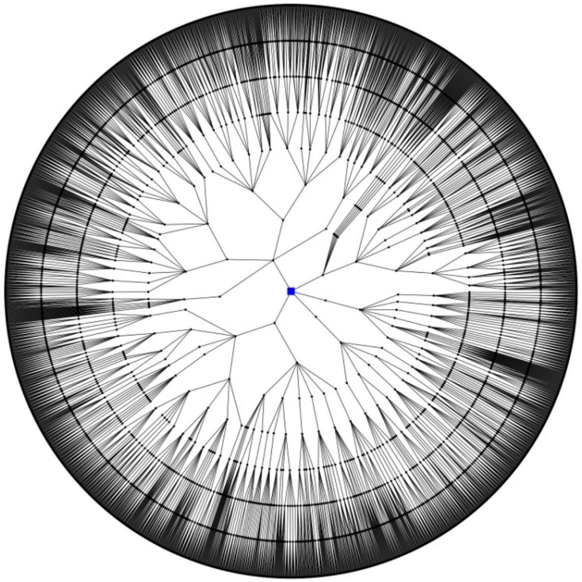
Two Search Philosophies

UCT Tree



- Asymmetric tree

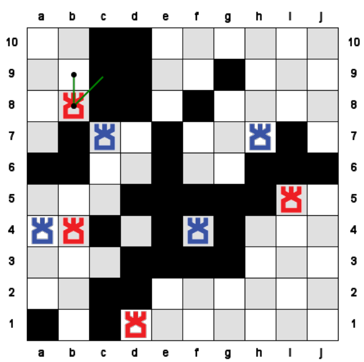
Minimax Tree

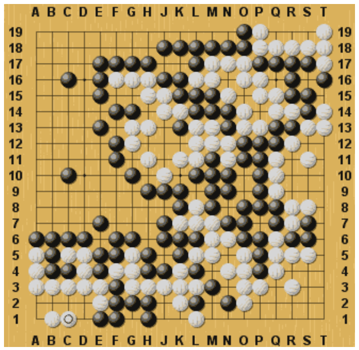


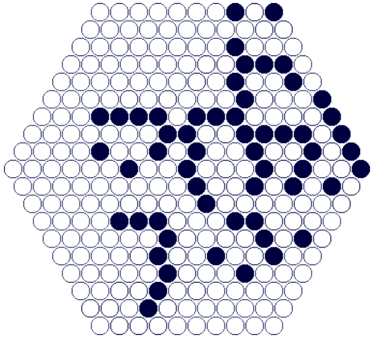
- Complete tree up to some depth bound

Two Search Philosophies

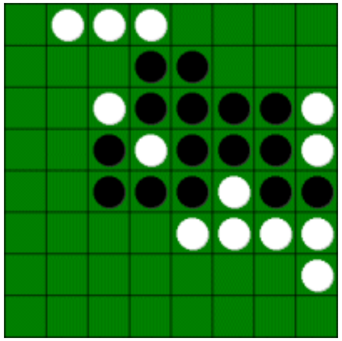
UCT



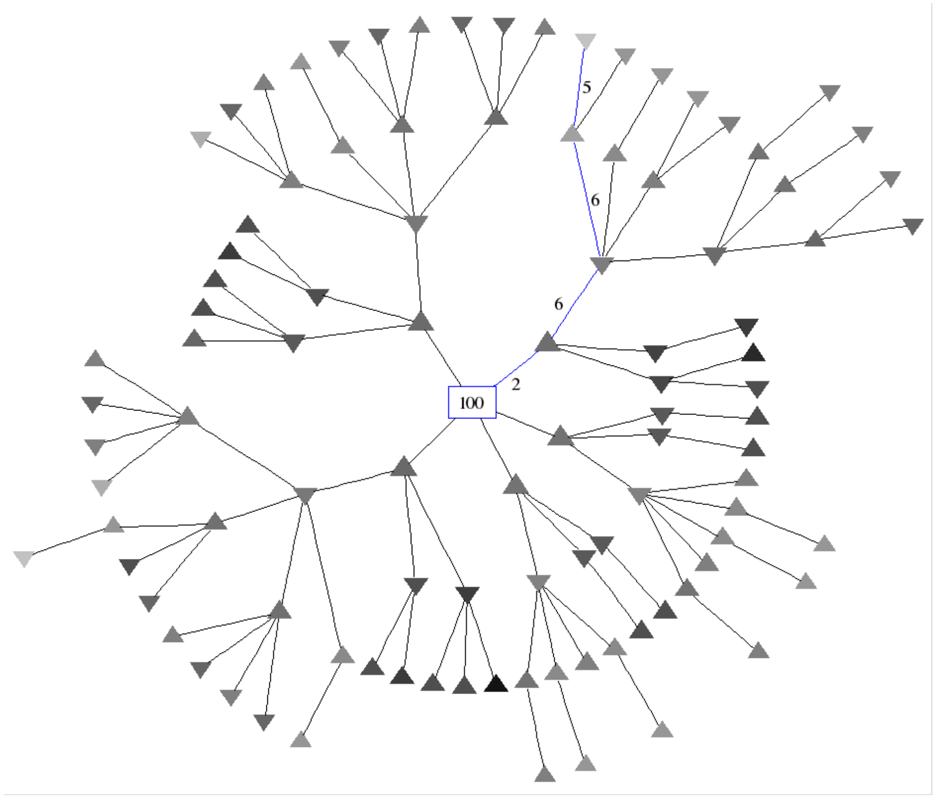




Minimax



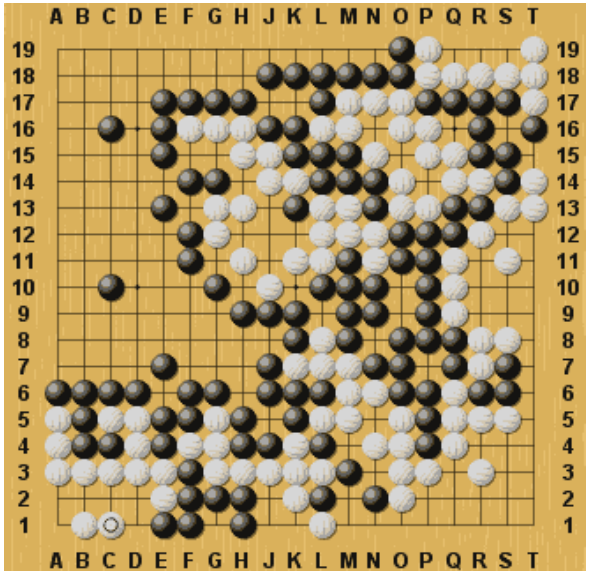
UCT in action



Why does UCT work in some domains but not others?

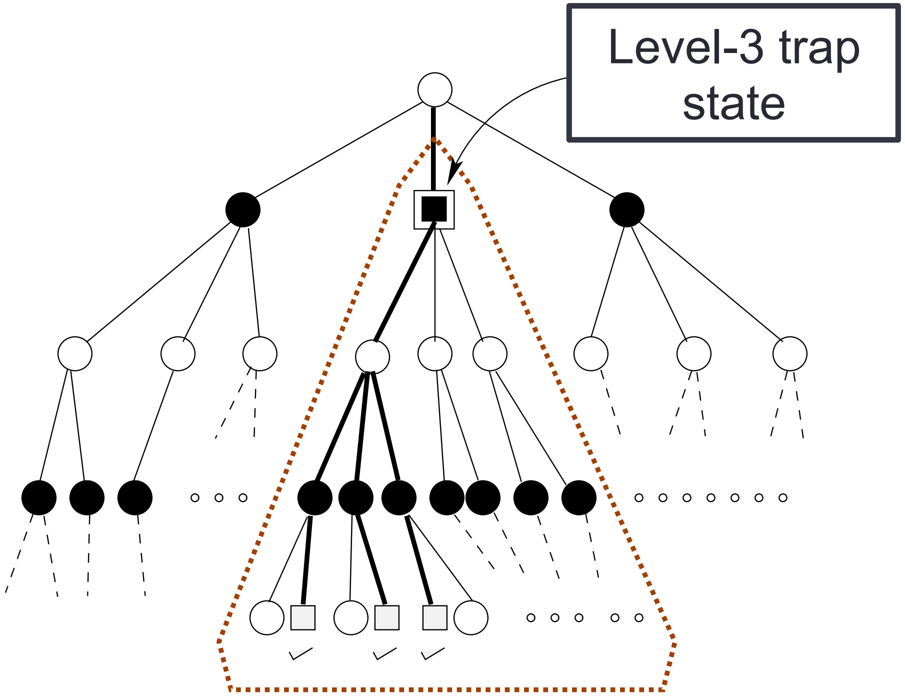
How is Chess different? Or, why just sampling of the game tree does not work?

Winning is defined by a small portion of the state



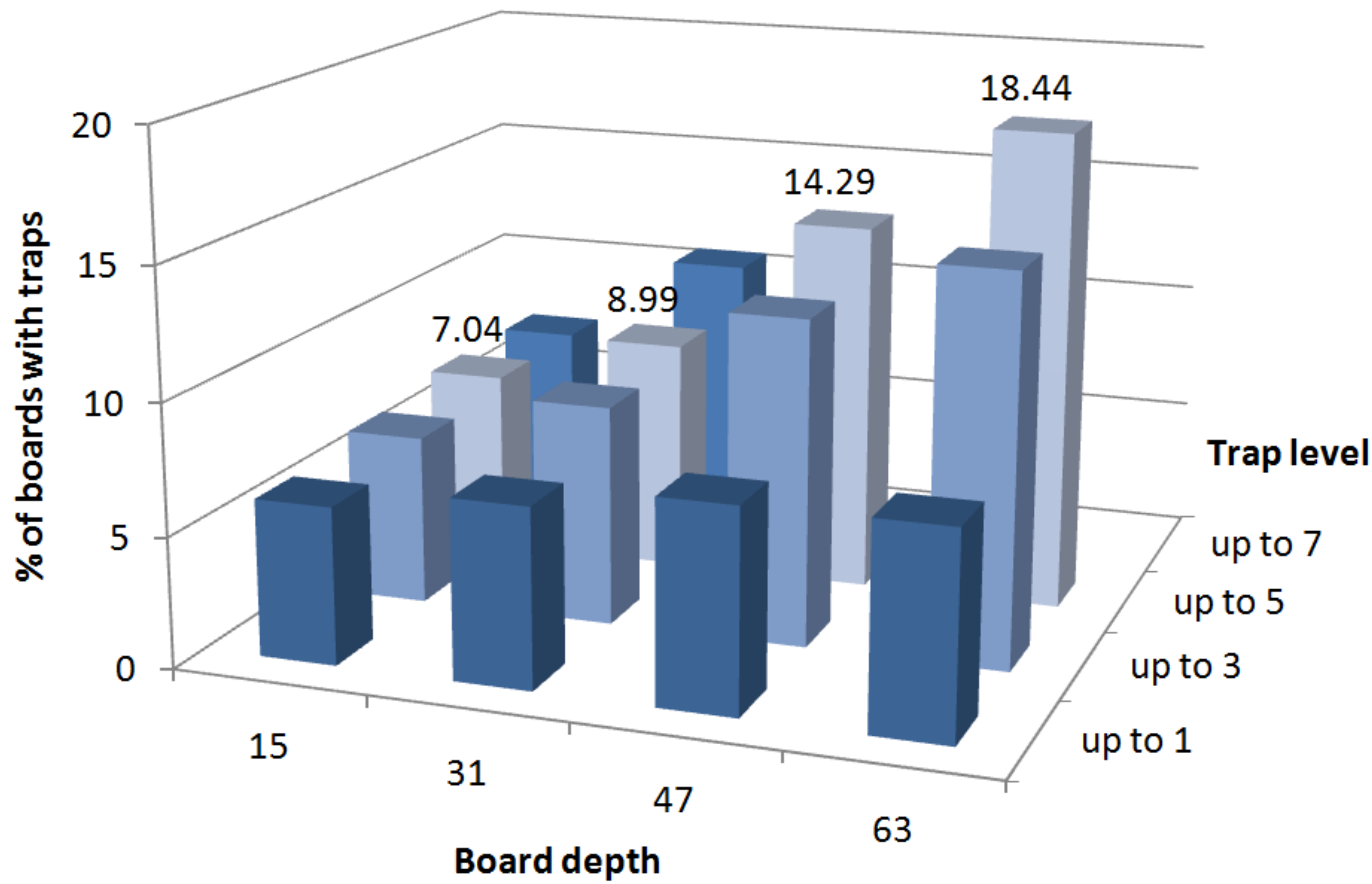
Winning is defined by a global function of the state

Trap States



Level-*k* search trap: position from where opponent can force a win in *k* steps (with optimal play)

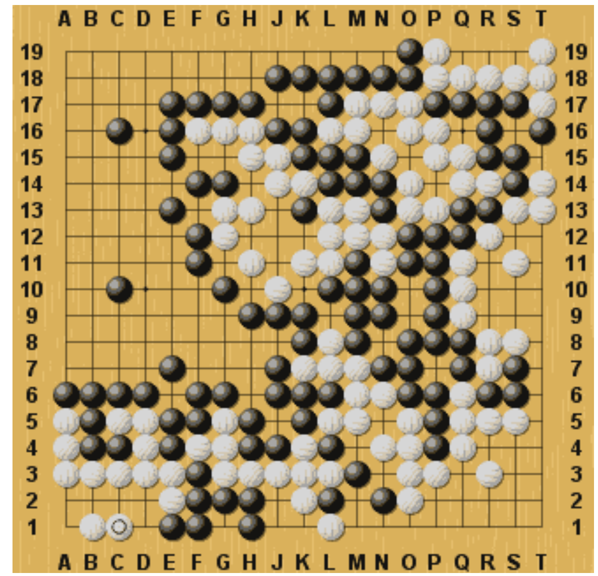
Shallow Trap States in Chess:
even in top-level games, “traps everywhere”



How is Chess different?

Shallow trap states are sprinkled throughout the search space

Sampling may miss these!!



Trap states only appear in the endgame

Summary

Game systems rely heavily on

- Search techniques
- Heuristic functions
- Bounding and pruning techniques
- Knowledge database on game

For AI, the abstract nature of games makes them an appealing subject for study:

state of the game is easy to represent;
agents are usually restricted to a small number of actions whose outcomes are defined by precise rules

Game playing was one of the first tasks undertaken in AI as soon as computers became programmable (e.g., Turing, Shannon, and Wiener tackled chess).

Game playing research has spawned a number of interesting research ideas on search, data structures, databases, heuristics, evaluations functions and other areas of computer science.