CSC 550: Introduction to Artificial Intelligence

Fall 2008

search in game playing

- zero-sum games
- game trees, minimax principle
- alpha-beta pruning
- recent developments & applications

1

Search in game playing

consider games involving:

- 2 players
- perfect information
- zero-sum (player's gain is opponent's loss)

examples: tic-tac-toe, checkers, chess, othello, …

non-examples: poker, backgammon, prisoner's dilemma, …

von Neumann (the father of game theory) showed that for such games, there is always a "rational" strategy

- that is, can always determine a best move, assuming the opponent is equally rational

		O
		X
	O	X

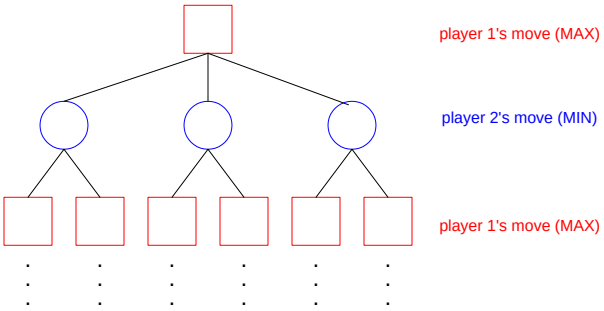
what is X's rational move?

2

Game trees

idea: model the game as a search tree

- associate a value with each game state (possible since zero-sum)
  player 1 wants to maximize the state value (call him/her MAX)
  player 2 wants to minimize the state value (call him/her MIN)
- players alternate turns, so differentiate MAX and MIN levels in the tree



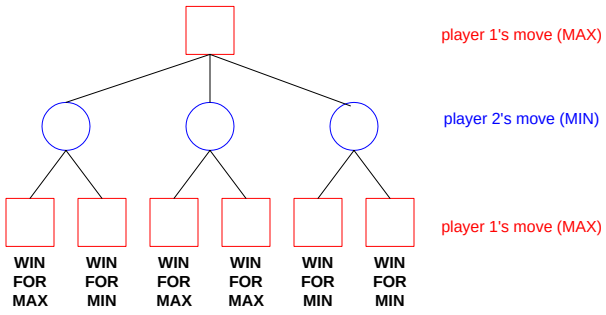
the leaves of the tree will be end-of-game states

3

Minimax search

minimax search:

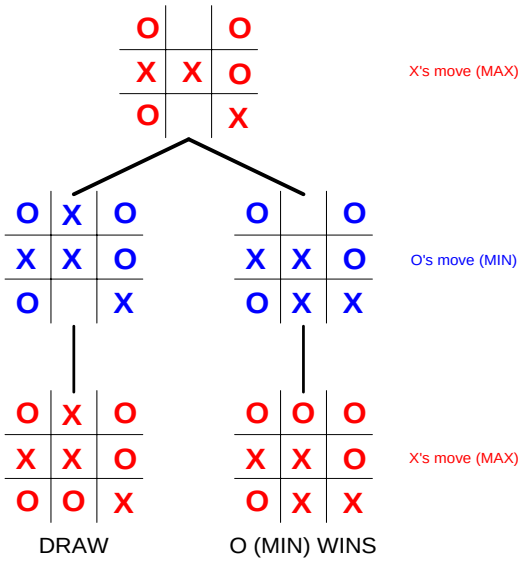
- at a MAX level, take the maximum of all possible moves
- at a MIN level, take the minimum of all possible moves



can visualize the search bottom-up (start at leaves, work up to root)
likewise, can search top-down using recursion

4

Minimax example



5

In-class exercise

6

Minimax in practice

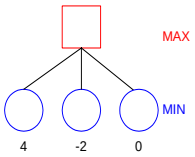
while Minimax Principle holds for all 2-party, perfect info, zero-sum games, an exhaustive search to find best move may be infeasible

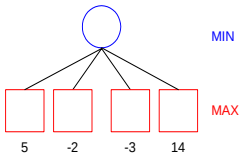
EXAMPLE: in an average chess game, ~100 moves with ~35 options/move
→ ~35^{100} states in the search tree!

practical alternative: limit the search depth and use heuristics

- expand the search tree a limited number of levels (limited look-ahead)
- evaluate the "pseudo-leaves" using a heuristic
  high value → good for MAX low value → good for MIN

back up the heuristic estimates to determine the best-looking move
at MAX level, take minimum at MIN level, take maximum



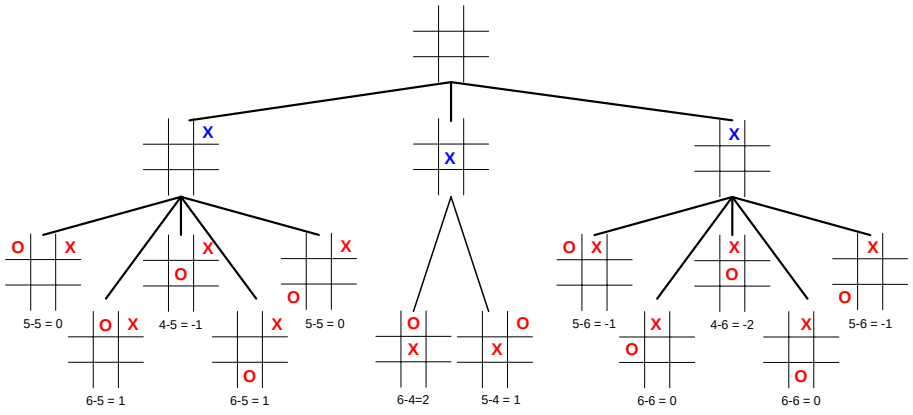


7

Tic-tac-toe example

heuristic(State) = {
- 1000 *if win for MAX (X)*
- -1000 *if win for MIN (O)*
- (#rows/cols/diags open for MAX – #rows/cols/diags open for MIN) *otherwise*

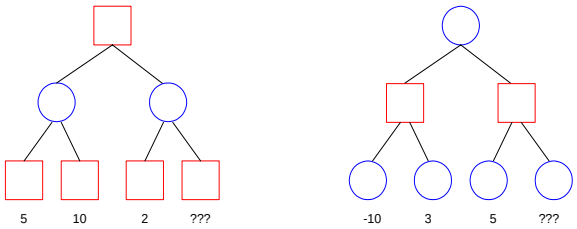
suppose look-ahead of 2 moves



8

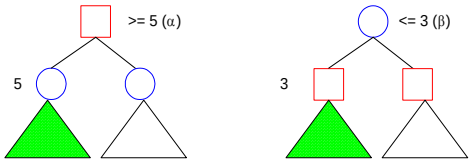
α-β bounds

sometimes, it isn't necessary to search the entire tree



α-β technique: associate bonds with state in the search

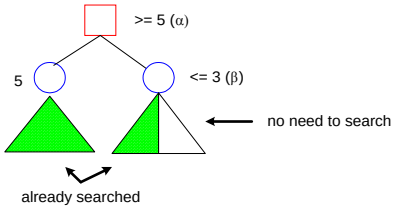
- associate lower bound α with MAX: can increase
- associate upper bound β with MIN: can decrease

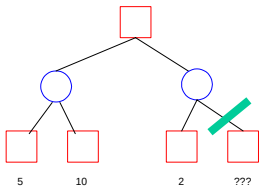


9

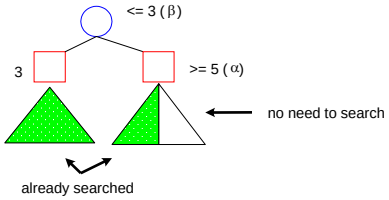
α-β pruning

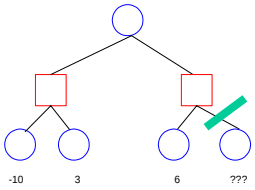
discontinue search below a MIN node if β value <= α value of ancestor





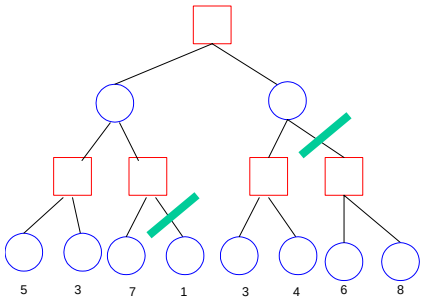
discontinue search below a MAX node if α value >= β value of ancestor





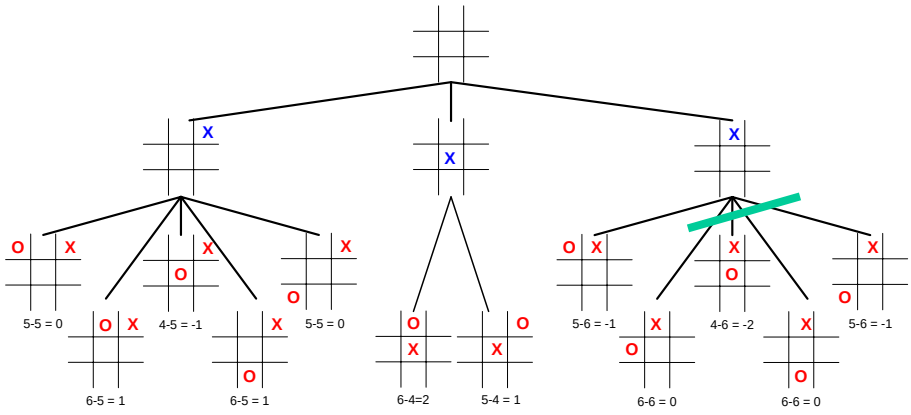
10

larger example



11

tic-tac-toe example



α-β vs. minimax:

worst case: α-β examines as many states as minimax

best case: assuming branching factor B and depth D, α-β examines ~$2b^{d/2}$ states

(i.e., as many as minimax on a tree with half the depth)

12

Articles to read & discuss

alpha-beta pruning – Wikipedia

Checkers is Solved – Jonathan Schaeffer et al., Science Magazine, 2007

Deep Blue – Murray Campbell, Joe Hoane & Feng-hsiung Hsu, 2001

AI Game-Playing Techniques: Are They Useful for Anything Other Than Games? – Dana Nau, IAAI, 1998

13