A quick review of search

Problem solving via search:
- To define the state space, define three things:
  - is_goal
  - next_states
  - starting_state

Uninformed search vs. informed search
- what’s the difference?
- what are the techniques we’ve seen?
- pluses and minuses?

Why should we study games?

Clear success criteria

Important historically for AI

Fun 😊

Good application of search
- hard problems (chess 35\(10^{19}\) states in search space, 10\(10^{40}\) legal states)

Some real-world problems fit this model
- game theory (economics)
- multi-agent problems

Adversarial Search

CS51A
David Kauchak
Spring 2019

Some material borrowed from:
Sara Owsley Sood and others
Types of games

What are some of the games you've played?

Types of games: game properties

- single-player vs. 2-player vs. multiplayer
- Fully observable (perfect information) vs. partially observable
- Discrete vs. continuous
- real-time vs. turn-based
- deterministic vs. non-deterministic (chance)

Strategic thinking = intelligence

For reasons previously stated, two-player games have been a focus of AI since its inception…

Important question: Is strategic thinking the same as intelligence?

Strategic thinking = intelligence

Humans and computers have different relative strengths in these games:

Humans

Computers
Strategic thinking = intelligence

Humans and computers have different relative strengths in these games:

- **Humans**: good at evaluating the strength of a board for a player
- **Computers**: good at looking ahead in the game to find winning combinations of moves

How could you figure out how humans approach playing chess?

An experiment was performed in which chess positions were shown to novice and expert players...

- Experts could reconstruct these perfectly
- Novice players did far worse...

Random chess positions (not legal ones) were then shown to the two groups...

Experts and novices did just as badly at reconstructing them!
People are still working on this problem…

http://people.brunel.ac.uk/~hsstffg/frg-research/chess_expertise/

Tic Tac Toe as search

If we want to write a program to play tic tac toe, what question are we trying to answer?

Given a state (i.e. board configuration), what move should we make!
Tic Tac Toe as search

How can we pose this as a search problem?

Eventually, we’ll get to a leaf

How does this help us?

Try and make moves that move us towards a win, i.e. where there are leaves with a WIN.
Minimizing risk

The computer doesn't know what move O (the opponent) will make.

It can assume that it will try and make the best move possible.

Even if O actually makes a different move, we're no worse off. Why?

Optimal Strategy

An **Optimal Strategy** is one that is at least as good as any other, no matter what the opponent does:

- If there's a way to force the win, it will.
- Will only lose if there's no other option.
Defining a scoring function

<table>
<thead>
<tr>
<th>WIN</th>
<th>TIE</th>
<th>LOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1</td>
<td>0</td>
<td>-1</td>
</tr>
</tbody>
</table>

Idea:
- define a function that gives us a “score” for how good each state is
- higher scores mean better

Our (X) turn

What should be the score of this state?

+1: we can get to a win

Opponent’s (O) turn

What should be the score of this state?

-1: we can get to a win
Defining a scoring function

Our (X) turn

What should be the score of this state?

0: If we play perfectly and so does O, the best we can do is a tie (could do better if O makes a mistake)
How can X play optimally?

Start from the leaves and propagate the score up:
- if X’s turn, pick the move that maximizes the utility
- if O’s turn, pick the move that minimizes the utility

Minimax Algorithm: An Optimal Strategy

\[
\text{minimax}(\text{state}) =
\begin{cases} 
\text{score}(\text{state}) & \text{if state is a terminal state} \\
\text{max}_{\text{s}} \text{minimax}(s) & \text{else if MY turn}
\end{cases}
\]

Uses recursion to compute the “value” of each state

Searches down to the leaves, then the values are “backed up” through the tree as the recursion finishes

What type of search is this?

What does this assume about how MIN will play? What if this isn’t true?
Baby Nim

Take 1 or 2 at each turn
Goal: take the last match

What move should I take?