Monte Carlo Tree Search

CS 227B 2019 – Week 5
Monte Carlo Search

Which path will MCS choose?
Monte Carlo Search

Which path will MCS choose?
Monte Carlo Tree Search

Which path(s) do we want to learn more about?
Monte Carlo Tree Search

Select → Expand → Simulate → Backpropagate
Single-Player MCTS
Step 1. Select

Exploration weight:
Spend time tuning this!

\[ UCT = \mu + C \sqrt{\frac{\log(n_{\text{parent}})}{n}} \]

Exploitation:
Prefer states with good average value

Exploration:
Prefer states we don’t know much about

\( \mu = \text{average reward from state} \)

\( n = \text{number of times state visited} \)
Step 1. Select

Exploration weight: Spend time tuning this!

\[ UCT = \mu + C \sqrt{\frac{\log(n_{parent})}{n}} \]

Exploitation: Prefer states with good average value

Exploration: Prefer states we don’t know much about

\( \mu = \text{average reward from state} \)
\( n = \text{number of times state visited} \)
Step 1. Select

**Exploration weight:**
Spend time tuning this!

\[ UCT = \mu + C \sqrt{\frac{\log(n_{parent})}{n}} \]

**Exploitation:**
Prefer states with good average value

**Exploration:**
Prefer states we don’t know much about

\( \mu = \text{average reward from state} \)
\( n = \text{number of times state visited} \)
Step 2. Expand

- $\mu=50$ $n=61$
- $\mu=60$ $n=10$
- $\mu=40$ $n=30$
- $\mu=60$ $n=20$
- $\mu=60$ $n=8$
- $\mu=50$ $n=1$
- $\mu=59$ $n=19$
- $\mu=\text{nan}$ $n=0$
- $\mu=\text{nan}$ $n=0$
Step 3. Simulate

- $\mu = 50$, $n = 61$
- $\mu = 60$, $n = 10$
- $\mu = 40$, $n = 30$
- $\mu = 60$, $n = 20$
- $\mu = 60$, $n = 8$
- $\mu = 50$, $n = 1$
- $\mu = \text{nan}$, $n = 0$
- $\mu = \text{nan}$, $n = 0$
- $\mu = 59$, $n = 19$
- $\mu = \text{nan}$, $n = 0$
- $\mu = \text{nan}$, $n = 0$
Step 3. Simulate

\[ \mu = 50 \quad n = 61 \]
\[ \mu = 60 \quad n = 10 \]
\[ \mu = 40 \quad n = 30 \]
\[ \mu = 60 \quad n = 20 \]
\[ \mu = 60 \quad n = 8 \]
\[ \mu = 59 \quad n = 19 \]
\[ \mu = \text{nan} \quad n = 0 \]
\[ \mu = \text{nan} \quad n = 0 \]

100
Step 4. Backpropagate

\[ \mu = 50 \quad n = 61 \]
\[ \mu = 60 \quad n = 10 \]
\[ \mu = 40 \quad n = 30 \]
\[ \mu = 60 \quad n = 20 \]
\[ \mu = 60 \quad n = 8 \]
\[ \mu = 50 \quad n = 1 \]
\[ \mu = \text{nan} \quad n = 0 \]
\[ \mu = \text{nan} \quad n = 0 \]

100
Step 4. Backpropagate

\[
\begin{align*}
\mu &= 50 \\
n &= 61 \\
\mu &= 60 \\
n &= 10 \\
\mu &= 40 \\
n &= 30 \\
\mu &= 60 \\
n &= 20 \\
\mu &= 60 \\
n &= 8 \\
\mu &= 50 \\
n &= 1 \\
\mu &= nan \\
n &= 0 \\
\mu &= 100 \\
n &= 1 \\
\mu &= nan \\
n &= 0 \\
\end{align*}
\]
Step 4. Backpropagate
Step 4. Backpropagate

\[
\begin{align*}
\mu &= 50 \\
n &= 61 \\
\mu &= 64 \\
n &= 11 \\
\mu &= 60 \\
n &= 8 \\
\mu &= 75 \\
n &= 2 \\
\mu &= 100 \\
n &= 1 \\
\mu &= \text{nan} \\
n &= 0 \\
\mu &= 59 \\
n &= 19 \\
\mu &= 60 \\
n &= 20 \\
\mu &= 50 \\
n &= 61 \\
\end{align*}
\]
Step 4. Backpropagate

- $\mu = 51$, $n = 62$
- $\mu = 64$, $n = 11$
  - $\mu = 60$, $n = 8$
  - $\mu = 75$, $n = 2$
    - $\mu = 100$, $n = 1$
      - $\mu = \text{nan}$, $n = 0$
    - $\mu = 60$, $n = 30$
  - $\mu = 60$, $n = 20$
- $\mu = 59$, $n = 19$
- $\mu = 59$, $n = 19$
- $\mu = 100$, $n = 1$
FAQ

• Which move should I pick at the end?
  • Either the one with the highest average reward, or the one that’s been visited the most (these will almost always be the same move).

• How many depth charges per state?
  • One, unless you have a good reason to do more.

• How should I pick the exploration weight?
  • Empirical testing. For a good starting value, think about confidence intervals.
  • The weight doesn’t have to be constant!

• What about multiplayer?
Multiplayer MCTS
Idea 1: Minimax

\[ \mu + C \sqrt{\frac{\log(n_{\text{parent}})}{n}} \]

pick child with max
**Idea 1: Minimax**

\[
\mu = 50 \\
n = 60
\]

\[
\mu = 60 \\
n = 10
\]

\[
\mu = 60 \\
n = 8
\]

\[
\mu = 60 \\
n = 1
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 50 \\
n = 10
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 40 \\
n = 30
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 60 \\
n = 20
\]

\[
\mu = 60 \\
n = 10
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 60 \\
n = 8
\]

\[
\mu = 60 \\
n = 1
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 60 \\
n = 8
\]

\[
\mu = 60 \\
n = 1
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 60 \\
n = 8
\]

\[
\mu = 60 \\
n = 1
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 60 \\
n = 8
\]

\[
\mu = 60 \\
n = 1
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 60 \\
n = 8
\]

\[
\mu = 60 \\
n = 1
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 60 \\
n = 8
\]

\[
\mu = 60 \\
n = 1
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 60 \\
n = 8
\]

\[
\mu = 60 \\
n = 1
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 60 \\
n = 8
\]

\[
\mu = 60 \\
n = 1
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 60 \\
n = 8
\]

\[
\mu = 60 \\
n = 1
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 60 \\
n = 8
\]

\[
\mu = 60 \\
n = 1
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 60 \\
n = 8
\]

\[
\mu = 60 \\
n = 1
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 60 \\
n = 8
\]

\[
\mu = 60 \\
n = 1
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 60 \\
n = 8
\]

\[
\mu = 60 \\
n = 1
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 60 \\
n = 8
\]

\[
\mu = 60 \\
n = 1
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 60 \\
n = 8
\]

\[
\mu = 60 \\
n = 1
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]

\[
\mu = 50 \\
n = 20
\]

\[
\mu = 30 \\
n = 10
\]
Idea 1: Minimax

\[
\mu - C \frac{\log(n_{\text{parent}})}{n}
\]

pick child with \textbf{min}

[Diagram showing a tree structure with nodes labeled \(\mu\) and \(n\), and state transitions indicated with arrows and labels X, Y, P, Q, R.]
Idea 2: One tree layer per role and timestep

\[ \mu_B = 60 \quad n = 10 \]

\[ \mu_R = 70 \quad n = 8 \]

\[ \mu_R = 50 \quad n = 1 \]

\[ \mu_B = 40 \quad n = 30 \]

\[ \mu_R = 80 \quad n = 10 \]

\[ \mu_R = 50 \quad n = 20 \]

\[ \mu_R = 50 \quad n = 20 \]

\[ \mu_R = 30 \quad n = 10 \]
Idea 3: One tree layer per timestep

moves=[P, Q, R]  
μ=[60, 40, 60]  
n=[1, 3, 2]

moves=[X, Y]  
μ=[60, 40]  
n=[3, 2]
Which variant should I use?

It’s up to you!
<table>
<thead>
<tr>
<th>Minimax</th>
<th>One Layer per Role</th>
<th>Merged Nodes</th>
</tr>
</thead>
</table>

More general.

Similar to things already seen.

worst case.

alpha-beta!
Extensions

• Merge identical states
  • Which direction(s) do you backpropagate?

• Incorporate knowledge of terminal state values
  • Can you find a way to incorporate an “alpha-beta pruning” equivalent?
  • …even without using minimax?

• Use multiple threads/processes
  • Where to parallelize?

• Your extension here
Questions?
Next topic: Arena