Game Playing
Part 2 Alpha-Beta Pruning

Yingyu Liang
yliang@cs.wisc.edu
Computer Sciences Department
University of Wisconsin, Madison

[based on slides from A. Moore http://www.cs.cmu.edu/~awm/tutorials, C. Dyer, J. Skrentny, Jerry Zhu]
alpha-beta pruning

Gives the same game theoretic values as minimax, but prunes part of the game tree.

"If you have an idea that is surely bad, don't take the time to see how truly awful it is." -- Pat Winston
• Depth-first order
• After returning from A, Max can get at least 100 at S
• After returning from F, Max can get at most 20 at B
• At this point, Max loses interest in B
• There is no need to explore G. The subtree at G is pruned. Saves time.
Alpha-beta pruning

function Max-Value(s, α, β)
inputs:
  s: current state in game, Max about to play
  α: best score (highest) for Max along path to s
  β: best score (lowest) for Min along path to s
output: \( \min(β, \text{best-score (for Max) available from } s) \)
  if (s is a terminal state)
  then return (terminal value of s)
  else for each \( s' \) in Succ(s)
    \( α := \max(α, \text{Min-value}(s',α,β)) \)
    if (\( α ≥ β \)) then return \( β \) /* alpha pruning */
  return \( α \)

Starting from the root:
Max-Value(root, -∞, +∞)
### Alpha-beta pruning

**Function** \( \text{Max-Value}(s, \alpha, \beta) \)

**Inputs:**
- \( s \): current state in game, Max about to play
- \( \alpha \): best score (highest) for Max along path to \( s \)
- \( \beta \): best score (lowest) for Min along path to \( s \)

**Output:** \( \min(\beta, \text{best-score (for Max) available from } s) \)

1. If \( s \) is a terminal state, return \( \text{terminal value of } s \)
2. Else for each \( s' \) in \( \text{Succ}(s) \)
   - \( \alpha := \max(\alpha, \text{Min-value}(s',\alpha,\beta)) \)
   - If \( \alpha \geq \beta \) then return \( \beta \) /* alpha pruning */
   - Return \( \alpha \)

**Function** \( \text{Min-Value}(s, \alpha, \beta) \)

**Output:** \( \max(\alpha, \text{best-score (for Min) available from } s) \)

1. If \( s \) is a terminal state, return \( \text{terminal value of } s \)
2. Else for each \( s' \) in \( \text{Succs}(s) \)
   - \( \beta := \min(\beta, \text{Max-value}(s',\alpha,\beta)) \)
   - If \( \alpha \geq \beta \) then return \( \alpha \) /* beta pruning */
   - Return \( \beta \)

Starting from the root:
Max-Value(root, -\( \infty \), +\( \infty \))
Alpha-beta pruning example 1

- Keep two bounds along the path
  - $\alpha$: the best Max can do
  - $\beta$: the best (smallest) Min can do
- If at anytime $\alpha$ exceeds $\beta$, the remaining children are pruned.
Alpha-beta pruning example 1

max

min

\[ \alpha = -\infty \]
\[ \beta = +\infty \]
Alpha-beta pruning example 1

max

min

\[ \alpha = -\infty \]
\[ \beta = +\infty \]

\[ \alpha = -\infty \]
\[ \beta = 200 \]
Alpha-beta pruning example 1

max

min

\[ \alpha = -\infty \]
\[ \beta = +\infty \]

\[ \alpha = -\infty \]
\[ \beta = 100 \]
Alpha-beta pruning example 1

max

min

\[ \alpha = -\infty \]
\[ \beta = 100 \]

\[ \alpha = 100 \]
\[ \beta = +\infty \]
Alpha-beta pruning example 1

\[
\begin{align*}
\text{max} & \\
\text{min} & \\
A & \rightarrow S \\
B & \rightarrow G \\
C & \rightarrow D \\
D & \rightarrow E \\
E & \rightarrow F \\
F & \rightarrow G \\
A & = 100 \\
C & = 200 \\
D & = 100 \\
E & = 120 \\
F & = 20 \\
S & = +\infty \\
B & = +\infty \\
\end{align*}
\]
**Alpha-beta pruning example 1**

The diagram illustrates a decision tree with nodes labeled with values and edges indicating the path taken. The values are labeled as follows:

- **S** with $\alpha = 100$ and $\beta = +\infty$
- **A** with $\alpha = -\infty$ and $\beta = 100$
- **B** with $\alpha = 100$ and $\beta = 120$
- **C** with value 200
- **D** with value 100
- **E** with value 120
- **F** with value 20
- **G**

The process of alpha-beta pruning involves updating the alpha ($\alpha$) and beta ($\beta$) values as the algorithm traverses the tree. The pruning occurs when the $\alpha$ value of the minimizer is greater than or equal to the $\beta$ value of the maximizer at any node, indicating that further exploration is unnecessary for that branch.
Alpha-beta pruning example 1

\[
\begin{align*}
\text{max} \quad & \alpha = 100 \\
\text{min} \quad & \alpha = -\infty \\
& \beta = 100 \\
& \beta = +\infty \\
& \alpha = 100 \\
& \beta = 20
\end{align*}
\]
**Alpha-beta pruning**

function **Max-Value** \((s, \alpha, \beta)\)  
inputs:  
s: current state in game, Max about to play  
\(\alpha\): best score (highest) for Max along path to \(s\)  
\(\beta\): best score (lowest) for Min along path to \(s\)  
output: \(\text{min}(\beta, \text{best-score (for Max) available from } s)\)  

if ( \(s\) is a terminal state )  
then return ( terminal value of \(s\) )  
else for each \(s'\) in Succ(s)  
\(\alpha := \text{max}(\alpha, \text{Min-value}(s', \alpha, \beta))\)  
if ( \(\alpha \geq \beta\) ) then return \(\beta\)  
/* alpha pruning */  
return \(\alpha\)

function **Min-Value** \((s, \alpha, \beta)\)  
output: \(\text{max}(\alpha, \text{best-score (for Min) available from } s)\)  

if ( \(s\) is a terminal state )  
then return ( terminal value of \(s\) )  
else for each \(s'\) in Succs(s)  
\(\beta := \text{min}(\beta, \text{Max-value}(s', \alpha, \beta))\)  
if (\(\alpha \geq \beta\)) then return \(\alpha\)  
/* beta pruning */  
return \(\beta\)

Starting from the root:  
Max-Value(root, \(-\infty, +\infty\)
Alpha-beta pruning example 2

What are the alpha and beta values on S?

• Keep two bounds along the path
  ▪ $\alpha$: the best Max can do
  ▪ $\beta$: the best (smallest) Min can do
• If at anytime $\alpha$ exceeds $\beta$, the remaining children are pruned.
Keep two bounds along the path
  ▪ $\alpha$: the best Max can do
  ▪ $\beta$: the best (smallest) Min can do
• If at anytime $\alpha$ exceeds $\beta$, the remaining children are pruned.
Alpha-beta pruning example 2

- Keep two bounds along the path
  - \( \alpha \): the best Max can do
  - \( \beta \): the best (smallest) Min can do
- If at anytime \( \alpha \) exceeds \( \beta \), the remaining children are pruned.
Alpha-beta pruning example 2

max

min

max

• Keep two bounds along the path
  ▪ $\alpha$: the best Max can do
  ▪ $\beta$: the best (smallest) Min can do
• If at anytime $\alpha$ exceeds $\beta$, the remaining children are pruned.
Keep two bounds along the path
- $\alpha$: the best Max can do
- $\beta$: the best (smallest) Min can do

If at anytime $\alpha$ exceeds $\beta$, the remaining children are pruned.
• Keep two bounds along the path
  ▪ \( \alpha \): the best Max can do
  ▪ \( \beta \): the best (smallest) Min can do
• If at anytime \( \alpha \) exceeds \( \beta \), the remaining children are pruned.
Alpha-beta pruning example 2

max

min

max

\( \alpha = 20 \)
\( \beta = +\infty \)

• Keep two bounds along the path
  ▪ \( \alpha \): the best Max can do
  ▪ \( \beta \): the best (smallest) Min can do

• If at anytime \( \alpha \) exceeds \( \beta \), the remaining children are pruned.
Alpha-beta pruning example 2

- Keep two bounds along the path
  - $\alpha$: the best Max can do
  - $\beta$: the best (smallest) Min can do
- If at anytime $\alpha$ exceeds $\beta$, the remaining children are pruned.

What are the alpha and beta values on A?
Alpha-beta pruning example 2

max

min

max

Keep two bounds along the path
- \( \alpha \): the best Max can do
- \( \beta \): the best (smallest) Min can do

If at anytime \( \alpha \) exceeds \( \beta \), the remaining children are pruned.
Alpha-beta pruning example 2

- Keep two bounds along the path
  - $\alpha$: the best Max can do
  - $\beta$: the best (smallest) Min can do
- If at anytime $\alpha$ exceeds $\beta$, the remaining children are pruned.
Alpha-beta pruning example 2

\[
\begin{align*}
\alpha & = -20 \\
\beta & = 20
\end{align*}
\]

• Keep two bounds along the path
  - \(\alpha\): the best Max can do
  - \(\beta\): the best (smallest) Min can do
• If at anytime \(\alpha\) exceeds \(\beta\), the remaining children are pruned.
Keep two bounds along the path
- \( \alpha \): the best Max can do
- \( \beta \): the best (smallest) Min can do

If at anytime \( \alpha \) exceeds \( \beta \), the remaining children are pruned.
Yet another alpha-beta pruning example

- Keep two bounds along the path
  - $\alpha$: the best Max can do on the path
  - $\beta$: the best (smallest) Min can do on the path
- If at anytime $\alpha$ exceeds $\beta$, the remaining children are pruned.

[Example from James Skrentny]
Alpha-beta pruning example

• Keep two bounds along the path
  ▪ $\alpha$: the best Max can do on the path
  ▪ $\beta$: the best (smallest) Min can do on the path
• If a max node exceeds $\beta$, it is pruned.
• If a min node goes below $\alpha$, it is pruned.

[Example from James Skrentny]
Alpha-beta pruning example

- Keep two bounds along the path
  - $\alpha$: the best Max can do on the path
  - $\beta$: the best (smallest) Min can do on the path
- If a max node exceeds $\beta$, it is pruned.
- If a min node goes below $\alpha$, it is pruned.

[Example from James Skrentny]
Alpha-beta pruning example

- Keep two bounds along the path
  - $\alpha$: the best Max can do on the path
  - $\beta$: the best (smallest) Min can do on the path
- If a max node exceeds $\beta$, it is pruned.
- If a min node goes below $\alpha$, it is pruned.

[Example from James Skrentny]
Alpha-beta pruning example

- Keep two bounds along the path
  - $\alpha$: the best Max can do on the path
  - $\beta$: the best (smallest) Min can do on the path
- If a max node exceeds $\beta$, it is pruned.
- If a min node goes below $\alpha$, it is pruned.

Example from James Skrentny

\[
\begin{align*}
\alpha &= -4 \\
\beta &= +4 \\
\end{align*}
\]
Alpha-beta pruning example

- Keep two bounds along the path
  - $\alpha$: the best Max can do on the path
  - $\beta$: the best (smallest) Min can do on the path
- If a max node exceeds $\beta$, it is pruned.
- If a min node goes below $\alpha$, it is pruned.
Alpha-beta pruning example

- Keep two bounds along the path
  - \( \alpha \): the best Max can do on the path
  - \( \beta \): the best (smallest) Min can do on the path
- If a max node exceeds \( \beta \), it is pruned.
- If a min node goes below \( \alpha \), it is pruned.

[Example from James Skrentny]
Alpha-beta pruning example

- Keep two bounds along the path
  - $\alpha$: the best Max can do on the path
  - $\beta$: the best (smallest) Min can do on the path
- If a max node exceeds $\beta$, it is pruned.
- If a min node goes below $\alpha$, it is pruned.

[Example from James Skrentny]
Alpha-beta pruning example

- Keep two bounds along the path
  - \( \alpha \): the best Max can do on the path
  - \( \beta \): the best (smallest) Min can do on the path
- If a max node exceeds \( \beta \), it is pruned.
- If a min node goes below \( \alpha \), it is pruned.

[Example from James Skrentny]
Alpha-beta pruning example

• Keep two bounds along the path
  - \( \alpha \): the best Max can do on the path
  - \( \beta \): the best (smallest) Min can do on the path

• If a max node exceeds \( \beta \), it is pruned.
• If a min node goes below \( \alpha \), it is pruned.

[Example from James Skrentny]
Alpha-beta pruning example

- Keep two bounds along the path
  - $\alpha$: the best Max can do on the path
  - $\beta$: the best (smallest) Min can do on the path
- If a max node exceeds $\beta$, it is pruned.
- If a min node goes below $\alpha$, it is pruned.

[Example from James Skrentny]
**Alpha-beta pruning example**

- Keep two bounds along the path
  - $\alpha$: the best Max can do on the path
  - $\beta$: the best (smallest) Min can do on the path
- If a max node exceeds $\beta$, it is pruned.
- If a min node goes below $\alpha$, it is pruned.

\[ \beta = -3 \]
\[ \alpha = 4 \]
Alpha-beta pruning example

• Keep two bounds along the path
  ▪ \( \alpha \): the best Max can do on the path
  ▪ \( \beta \): the best (smallest) Min can do on the path

• If a max node exceeds \( \beta \), it is pruned.
• If a min node goes below \( \alpha \), it is pruned.
Alpha-beta pruning example

• Keep two bounds along the path
  ▪ \( \alpha \): the best Max can do on the path
  ▪ \( \beta \): the best (smallest) Min can do on the path

• If a max node exceeds \( \beta \), it is pruned.
• If a min node goes below \( \alpha \), it is pruned.

[Example from James Skrentny]

[Diagram of alpha-beta pruning example]
Alpha-beta pruning example

- Keep two bounds along the path
  - \( \alpha \): the best Max can do on the path
  - \( \beta \): the best (smallest) Min can do on the path
- If a max node exceeds \( \beta \), it is pruned.
- If a min node goes below \( \alpha \), it is pruned.

[Example from James Skrentny]
Alpha-beta pruning example

- Keep two bounds along the path
  - $\alpha$: the best Max can do on the path
  - $\beta$: the best (smallest) Min can do on the path
- If a max node exceeds $\beta$, it is pruned.
- If a min node goes below $\alpha$, it is pruned.

[Example from James Skrentny]
Alpha-beta pruning example

- Keep two bounds along the path
  - \( \alpha \): the best Max can do on the path
  - \( \beta \): the best (smallest) Min can do on the path

- If a max node exceeds \( \beta \), it is pruned.
- If a min node goes below \( \alpha \), it is pruned.

[Example from James Skrentny]
Alpha-beta pruning example

- Keep two bounds along the path
  - $\alpha$: the best Max can do on the path
  - $\beta$: the best (smallest) Min can do on the path
- If a max node exceeds $\beta$, it is pruned.
- If a min node goes below $\alpha$, it is pruned.

[Example from James Skrentny]
**Alpha-beta pruning example**

- Keep two bounds along the path
  - $\alpha$: the best Max can do on the path
  - $\beta$: the best (smallest) Min can do on the path
- If a max node exceeds $\beta$, it is pruned.
- If a min node goes below $\alpha$, it is pruned.

---

[Example from James Skrentny]
**Alpha-beta pruning example**

- Keep two bounds along the path
  - $\alpha$: the best Max can do on the path
  - $\beta$: the best (smallest) Min can do on the path
- If a max node exceeds $\beta$, it is pruned.
- If a min node goes below $\alpha$, it is pruned.

[Example from James Skrentny]
Alpha-beta pruning example

• Keep two bounds along the path
  ▪ $\alpha$: the best Max can do on the path
  ▪ $\beta$: the best (smallest) Min can do on the path
• If a max node exceeds $\beta$, it is pruned.
• If a min node goes below $\alpha$, it is pruned.

[Example from James Skrentny]
Alpha-beta pruning example

- Keep two bounds along the path
  - $\alpha$: the best Max can do on the path
  - $\beta$: the best (smallest) Min can do on the path
- If a max node exceeds $\beta$, it is pruned.
- If a min node goes below $\alpha$, it is pruned.

[Example from James Skrentny]
**Alpha-beta pruning example**

- Keep two bounds along the path
  - $\alpha$: the best Max can do on the path
  - $\beta$: the best (smallest) Min can do on the path
- If a max node exceeds $\beta$, it is pruned.
- If a min node goes below $\alpha$, it is pruned.

[Example from James Skrentny]
Alpha-beta pruning example

- Keep two bounds along the path
  - $\alpha$: the best Max can do on the path
  - $\beta$: the best (smallest) Min can do on the path
- If a max node exceeds $\beta$, it is pruned.
- If a min node goes below $\alpha$, it is pruned.

[Example from James Skrentny]
How effective is alpha-beta pruning?

- Depends on the order of successors!

- In the best case, the number of nodes to search is $O(b^{m/2})$, the square root of minimax’s cost.
- This occurs when each player's best move is the leftmost child.
- In DeepBlue (IBM Chess), the average branching factor was about 6 with alpha-beta instead of 35-40 without.
- The worst case is no pruning at all.