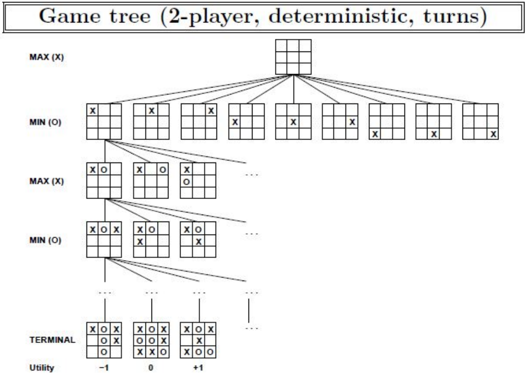
Introduction to Artificial Intelligence and Problem Formulations

Dr. Tapas Kumar Mishra

Assistant Professor
Department of Computer Science and Engineering
SRM University
Amaravati-522502, Andhra Pradesh, India
Email: tapaskumar.m [at] srmap [dot] edu [dot] in

Game Tree - Tic-Tac-Toe



- From the initial state, MAX has **nine** possible moves.
- 9! = 362880 terminal nodes
- Play alternates between MAX'S placing an X and MIN'S placing an 0 until we reach leaf nodes corresponding to terminal states such that one player has **three in a row** or all the **squares are filled**.

Different functions

- S_0: The **initial state**, which specifies how the game is set up at the start.
- PLAYER(s): Defines which player has the move in a state.
- ACTIONS(s): Returns the set of legal moves in a state.
- RESULT(s, a): The **transition model**, which defines the result of a move.
- TERMINAL-TEST(s): A **terminal test**, which is true when the game is over and false otherwise. States where the game has ended are called **terminal states**.
- UTILITY(s, p): A **utility function** (also called an objective function or payoff function), defines the final numeric value for a game that ends in terminal state s for a player p. In chess, the outcome is a win, loss, or draw, with values $+1$, 0, or $\frac{1}{2}$. Some games have a wider variety of possible outcomes; the payoffs in backgammon range from 0 to $+192$. A **zero-sum game** is (confusingly) defined as one where the total payoff to all players is the same for every instance of the game. Chess is zero-sum because every game has payoff of either $0 + 1$, $1 + 0$ or $\frac{1}{2} + \frac{1}{2}$. "Constant-sum" would have been a better term, but zero-sum is traditional and makes sense if you imagine each player is charged an entry fee of $\frac{1}{2}$.

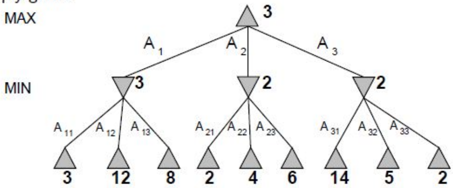
Optimal Decisions in Games - The minimax Algorithm

Minimax

Perfect play for deterministic, perfect-information games

Idea: choose move to position with highest minimax value
= best achievable payoff against best play

E.g., 2-ply game:



- MAX: Upper triangle and MIN: Lower triangle
- MAX'S best move at the root is A_1, it leads to the highest minimax value and MIN'S best reply is A_{11}, because it leads to the successor with the lowest minimax value.

Optimal Decisions in Games - The minimax Algorithm

- Given a game tree, the optimal strategy can be determined by examining the **minimax value** of each node, which we write as MINIMAX(n).

$$\text{MINIMAX-VALUE}(n) = \begin{cases} \text{UTILITY}(n) & \text{if n is a terminal state} \\ \max_{s \in Successors(n)} \text{MINIMAX-VALUE}(s) & \text{if } n \text{ is a MAX node} \\ \min_{s \in Successors(n)} \text{MINIMAX-VALUE}(s) & \text{if } n \text{ is a MIN node.} \end{cases}$$

Properties of minimax

Complete?? Yes, if tree is finite (chess has specific rules for this)

Optimal?? Yes, against an optimal opponent. Otherwise??

Time complexity?? $O(b^m)$

Space complexity?? $O(bm)$ (depth-first exploration)

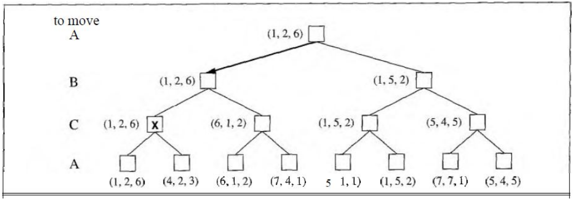
For chess, $b \approx 35$, $m \approx 100$ for "reasonable" games
$\Rightarrow$ exact solution completely infeasible

But do we need to explore every path?

- The first MIN node, has three successors with values 3, 12, and 8, so its minimax value is 3.
- The root node is a MAX node; its successors have minimax values 3, 2, and 2; so it has a minimax value of 3.

Optimal Decisions in Multiplayer Games

- Three Players: A, B, C, Vector: $<v_A, v_B, v_C>$
- (1, 2, **6**) and (4, 2, **3**) = (1, 2, 6) (Player C)



Optimal Decisions in Multiplayer Games

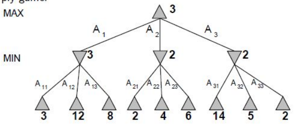
- Suppose A and B are in **weak positions** and C is in a **stronger position**.
- Then it is often **optimal** for **both A and B** to **attack C** rather than each other.
- Let **C** destroy each of them **individually**.
- In this way, collaboration emerges from purely **selfish behavior**. Of course, as soon as **C weakens** under the **joint onslaught**, the **alliance loses** its value, and **either A or B** could **violate** the **agreement**.

Minimax

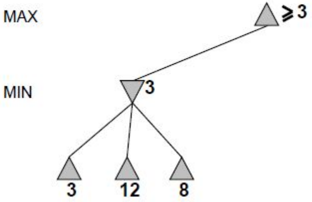
Perfect play for deterministic, perfect-information games

Idea: choose move to position with highest minimax value
= best achievable payoff against best play

E.g., 2-ply game:



α–β pruning example



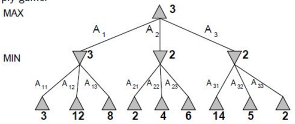
Alpha-Beta Pruning

Minimax

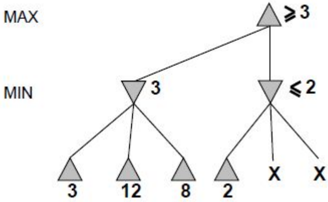
Perfect play for deterministic, perfect-information games

Idea: choose move to position with highest minimax value
= best achievable payoff against best play

E.g., 2-ply game:



α–β pruning example



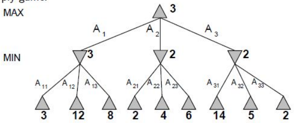
Alpha-Beta Pruning

Minimax

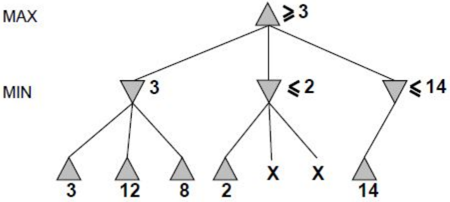
Perfect play for deterministic, perfect-information games

Idea: choose move to position with highest minimax value
= best achievable payoff against best play

E.g., 2-ply game:



α–β pruning example

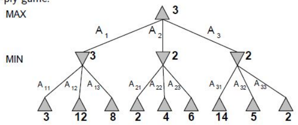


Minimax

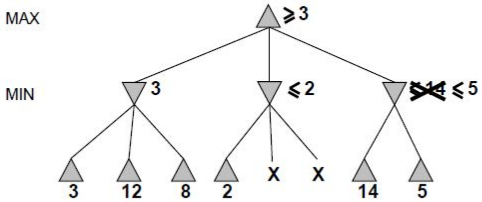
Perfect play for deterministic, perfect-information games

Idea: choose move to position with highest minimax value
= best achievable payoff against best play

E.g., 2-ply game:



α–β pruning example

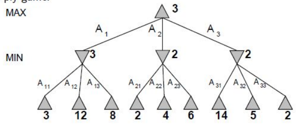


Minimax

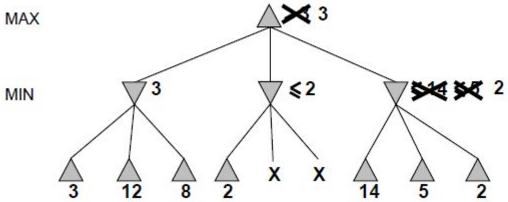
Perfect play for deterministic, perfect-information games

Idea: choose move to position with highest minimax value
= best achievable payoff against best play

E.g., 2-ply game:



α–β pruning example

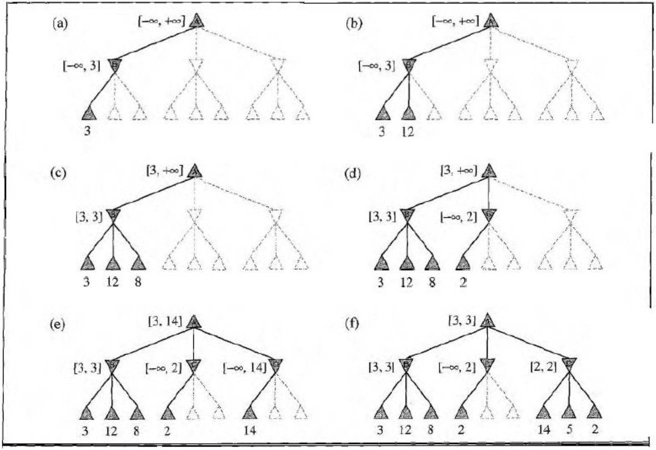


- When we apply **alpha-beta pruning** to a **standard minimax tree**, it returns the **same move** as **minimax** would, but **prunes** away **branches** that **cannot** possibly **influence** the **final decision**.

Example

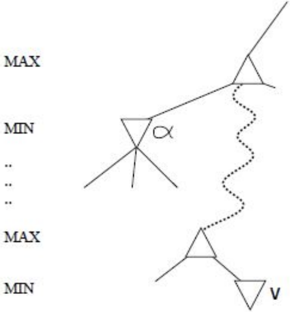
$$\text{MINIMAX(root)} = \max(\min(3, 12, 8), \min(2, x, y), \min(14, 5, 2))$$
$$= \max(3, \min(2, x, y), 2)$$
$$= \max(3, z, 2) \text{ where } z = \min(2, x, y) \leq 2$$
$$= 3$$

Alpha-Beta Pruning



- [at least, at most]

Why is it called α–β?



α is the best value (to MAX) found so far off the current path

If V is worse than α, MAX will avoid it $\Rightarrow$ prune that branch

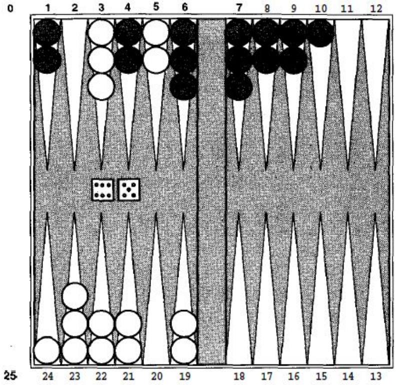
Define β similarly for MIN

Alpha-Beta Pruning

- Alpha-beta needs to examine only $O(b^{m/2})$ nodes to pick the best move, instead of $O(b^m)$ for minimax.
- This means that the effective branching factor becomes $\sqrt{b}$ instead of b-for chess, about 6 instead of 35.

Games that Include an Element of Chance

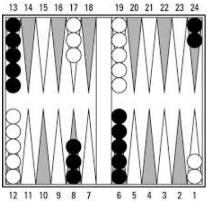
- **Backgammon** is a typical game that combines **luck** and **skill**.
- Two players, 24 narrow triangles, 15 checkers (white/black) per player
- 0 $\sim$ 6: Black's home board, 7 $\sim$ 12: outer board
- 19 $\sim$ 25: White's home board, 13 $\sim$ 18: outer board



Games that Include an Element of Chance

- The aim of the game is to move all one's pieces off the board.
- **White moves** clockwise toward **25**, and **black moves** counterclockwise toward **0**.
- A piece can **move to any position except one** where **there are two or more of the opponent's pieces**.
- If it **moves to a position with one opponent piece**, that piece is captured and has to **start its journey again from the beginning**.
- In this position, **white has just rolled 6-5** and has four legal moves: (5-10,5-11), (5-11,19-24), (5-10,10-16), and (5-11,11-16).

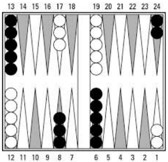
Games that Include an Element of Chance



Initial arrangement of checkers

- **1st player:** 2 on first, 5 on twelve and ninteen, and 3 on seventeen point.
- **2nd player:** 2 on twenty-fourth point, 5 on thirteen and sixth point, and 3 on eigth point.
- Both players have their own pair of dice and a dice cup used for shaking.
- To start the game, each player throws a single die. If equal numbers come up then both players roll again until they roll different numbers.
- The player throwing the higher number now moves his checkers according to the numbers showing on both dice.
- After the first roll, the players throw two dice and alternate turns.

Games that Include an Element of Chance



Players begin with two checkers on their 24-point, three checkers on their 8-point, and five checkers each on their 13-point and their 6-point.

Rules

- A checker may be moved only to an open point one, i.e., not occupied by 2 or more opposing checkers.
- The numbers on the 2 dice constitute separate moves. For example, if a player rolls 5 and 3, he may move one checker 5 spaces to an open point and another checker 3 spaces to an open point or he may move the one checker a total of 8 spaces to an open point, but only if the intermediate point (either 3 or 5 spaces from the starting point) is also open.

Means-end Analysis

Means-end Analysis

1. Comparing the current state S_i to a goal state S_g and computing the difference D_{ig}
2. An operator O_k is then selected to reduce the difference D_{ig}.
3. The operator O_k is applied if possible. If not, the current state is saved, a subgoal is created and means-end analysis is applied recursively to reduce the subgoal.
4. If the subgoal is solved, the saved state is restored and work is resumed on the original problem.

Example

Let the initial PL object $L_i = R \ \& \ (\sim P \rightarrow Q)$ and goal object $L_g = (Q \vee P) \ \& \ R$

$L_i^{'} = (\sim P \rightarrow Q) \ \& \ R = (\sim\sim P \vee Q) \ \& \ R = (P \vee Q) \ \& \ R$

$L_g = (Q \vee P) \ \& \ R$

NOTE: A comparison of the difference that R is on the left in L_i but on the right in L_g. This causes a subgoal to be set up to reduce this difference.

Thank You!