Discussion 6
Adversarial search and midterm review

Shirley Chen

11/01/2019

Game Search

Games:

- Require to make some decision when calculating the optimal decision is infeasible
- Penalize inefficiency severely

How to choose a good move when time is limited?

Game Search

- Pruning
  - Ignore portions of the search tree that make no difference to the final choice
- Evaluation functions
  - approximate the true utility of a state without doing a complete search

Games with Two Players

- S_0: The **initial state**, which specifies how the game is set up at the start.
- $\text{PLAYER}(s)$: Defines which player has the move in a state.
- $\text{ACTIONS}(s)$: Returns the set of legal moves in a state.
- $\text{RESULT}(s, a)$: The **transition model**, which defines the result of a move.
- $\text{TERMINAL-TEST}(s)$: A **terminal test**, which is true when the game is over and false otherwise. States where the game has ended are called **terminal states**.
- $\text{UTILITY}(s, p)$: 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

What is an optimal solution in adversarial search?

- Normal search:
  - a sequence of actions leading to a goal state
- Adversarial search:
  - Find a contingent strategy
    - First move, moves in the states resulting from the other guy's possible moves, ...

Optimal Decisions - MINIMAX

Given a game tree, how to determine the optimal strategy?

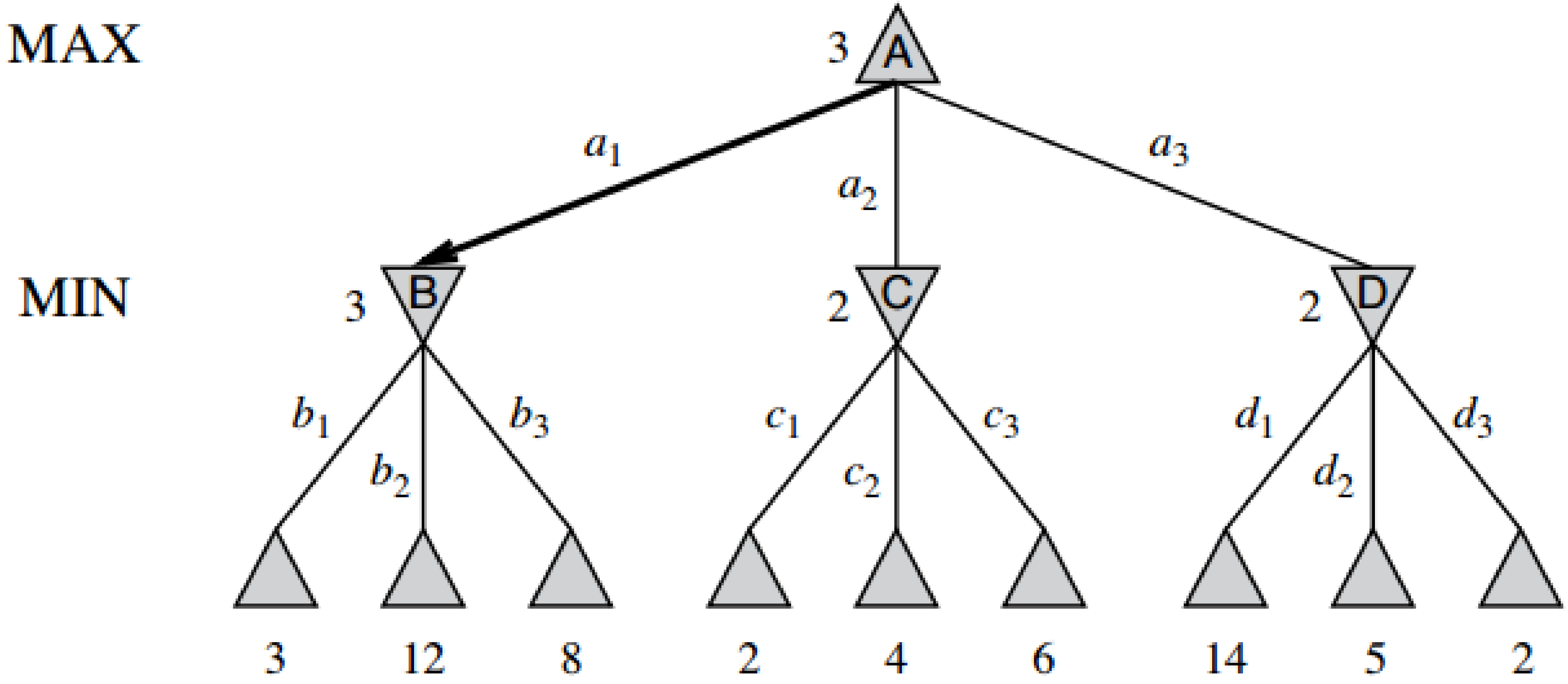
MINIMAX(n)

- The utility (for MAX)
- Assume both players play *optimally* from there to end of game
  - Given a choice, MAX prefers to move to a state of maximum value, whereas MIN prefers a state of minimum value.

$$\textsc{Minimax}(s) =$$

$$\begin{cases} \textsc{Utility}(s) & \text{if } \textsc{Terminal-Test}(s) \\ \max_{a \in Actions(s)} \textsc{Minimax}(\textsc{Result}(s,a)) & \text{if } \textsc{Player}(s) = \textsc{max} \\ \min_{a \in Actions(s)} \textsc{Minimax}(\textsc{Result}(s,a)) & \text{if } \textsc{Player}(s) = \textsc{min} \end{cases}$$

Optimal Decisions



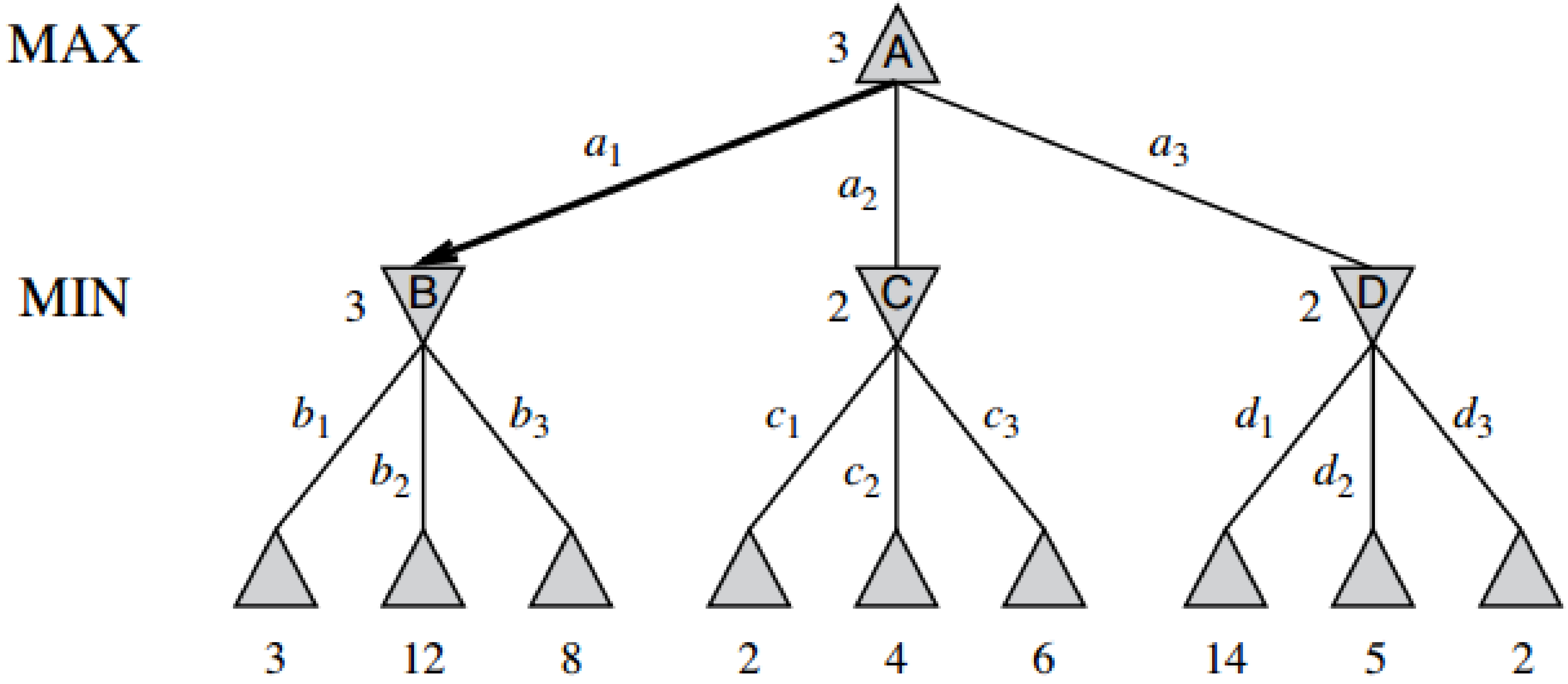
Alpha-beta pruning

- Minimax: a way of finding an optimal move in a two player game.
- Alpha-beta pruning: finding the optimal minimax solution while avoiding searching subtrees of moves which won't be selected.

- Alpha: maximum lower bound of possible solutions
- Beta: minimum upper bound of possible solutions

$$\alpha \leq N \leq \beta$$

Example



Alpha-beta pruning