Artificial Intelligence

Game Playing

MONTANA STATE UNIVERSITY

1

Mountains & Minds

Game Playing as Search

- State-space search assumes single agent searching in a fixed environment
- Game search introduces one or more additional agents
- Actions of all agents affect the environment
- For our discussions, we will assume only two players

MONTANA STATE UNIVERSITY

2

Mountains & Minds

Games vs. Search Problems

- “Unpredictable” opponent means solution is a “contingency plan.”
- Time limits restrict ability to search for goal prior to play—need to approximate.
- Plan of attack
  - Algorithm for perfect play (von Neuman)
  - Finite horizon, approximate evaluation (Shannon, Samuel)
  - Pruning to reduce cost (McCarthy)

MONTANA STATE UNIVERSITY

3

Mountains & Minds

Types of Games

- Perfect information games
  - Complete, unambiguous state description.
  - Complete, predictable knowledge of result of actions.
  - Assumes players alternate moves.
  - Examples: chess, checkers, othello.
- Imperfect information games
  - Ambiguity in state description.
  - Non-determinism in actions
  - Presence of chance player
  - Examples: poker, backgammon

Rational Play?

The Princess Bride

Non-Cooperative Games

- ***Def:*** A *non-cooperative game* is given by
  - A non-empty set, P, of players
  - The non-empty strategy spaces of the individual players $\{S_p \mid p \in P\}$
  - The set of real-valued payoff functions of the individual players $\{f_p \mid p \in P\}$ where

$$f_p : S \to \Re$$

$$S = \prod_{\forall p \in P} S_p$$

Non-Cooperative Games

- *P* is a set of players, typically of size 2.
- S_p is the strategy space (i.e., the set of *pure* strategies) for player *p*.
- *S* is the situation space for the game.
- A situation is the set of strategies, one for each player, in a play in the game.
- The function $f_p(s)$ is the payoff to player *p* given situation *s*.

MONTANA STATE UNIVERSITY

Constant-Sum Games

- A game is *constant-sum* if there exists a constant, $c \in \Re$, such that

$$\sum_{\forall p \in P} f_p(s) = c$$

- When $c = 0$, the game is a called a *zero-sum* game.
- All constant-sum games can be made into zero-sum games by factoring out *c*.

MONTANA STATE UNIVERSITY

Games in Normal Form

Bimatrix Game		
p_1/p_2	$s_{2,1}$	$s_{2,2}$
$s_{1,1}$	$f_1(s_{1,1},s_{2,1})/f_2(s_{1,1},s_{2,1})$	$f_1(s_{1,1},s_{2,2})/f_2(s_{1,1},s_{2,2})$
$s_{1,2}$	$f_1(s_{1,2},s_{2,1})/f_2(s_{1,2},s_{2,1})$	$f_1(s_{1,2},s_{2,2})/f_2(s_{1,2},s_{2,2})$
Matrix Game		
p_1/p_2	$s_{2,1}$	$s_{2,2}$
$s_{1,1}$	$f_1(s_{1,1},s_{2,1})$	$f_1(s_{1,1},s_{2,2})$
$s_{1,2}$	$f_1(s_{1,2},s_{2,1})$	$f_1(s_{1,2},s_{2,2})$

MONTANA STATE UNIVERSITY

Prisoner's Dilemma

- Two crooks are caught by the police.
- They are immediately segregated and interrogated.
- The police want a confession on a serious charge.
- They have evidence to convict each on a lesser charge.
- Each is approached with the same proposal.

MONTANA STATE UNIVERSITY

10

Mountains & Minds

The Proposal

- Conviction on the lesser charge will bring x years in prison.
- Conviction on the greater charge will bring $y > x$ years in prison.
- The prisoner can confess and implicate the other.
- If one confesses, and the other doesn't the one that cooperates will spend $z < x$ years in prison—the other gets y years.
- If both confess, then they each get w years in prison, where $x < w < y$.
- What should the prisoner do?

MONTANA STATE UNIVERSITY

11

Mountains & Minds

Prisoner's Dilemma

	Cooperate (Deny)	Defect (Confess)
Cooperate (Deny)	2/2	10/1
Defect (Confess)	1/10	5/5

MONTANA STATE UNIVERSITY

12

Mountains & Minds

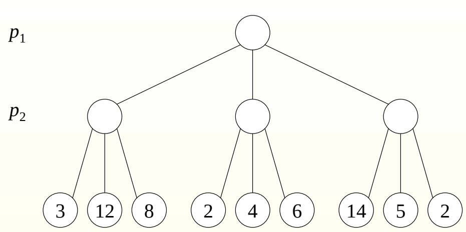
Stone-Paper-Scissors

	Stone	Paper	Scissors
Stone	0	–1	1
Paper	1	0	–1
Scissors	–1	1	0

Extensive-Form Games

- Suppose players alternative moves.
- Represent the game as a tree with each node corresponding to a state in the game.
- Apply the payoff at the leaves (usually)
- Each level of the tree is a *ply*.
- Examples:
  - Tic-Tac-Toe, Chess, Checkers

Game Trees



Standard Form

- ***Claim***: All extensive form games can be converted to normal form games.
- How?
- Practical?

Solving Games

A Beautiful Mind

Solving Games

- The objective of search in game playing is to determine strategies such that payoff is maximized.
- ***Def***: For any player p, any strategy $d_p \in S_p$, and any situation $s = \{s_q \mid q \in P\}$, let $s||d_p$ (called the *unilateral defection* from s by player p) denote the member of S obtained by replacing the p^{th} component of s by d_p.

Nash Equilibrium Point

- ***Def*:** Situation $s \in S$ is *admissible* for $p \in P$ (denoted $s \in A_p(\Gamma)$) if for all $d_p \in S_p$, we have $f_p\,(s||d_p) \le f_p(s)$. In other words, s is admissible for p if and only if p has no profitable unilateral defection from s.
- ***Def*:** Situation s is a *Nash equilibrium point* for Γ if it is admissible for *all* players, i.e., $EP(\Gamma) = \bigcap_{p \in P} A_p(\Gamma)$.
- What if $EP(\Gamma) = \varnothing$?

MONTANA STATE UNIVERSITY

19

Mountains & Minds

Mixed Strategies

- Let us enlarge the situation space, S, by expanding the strategy spaces S_p.
- Consider $\sigma_p \in \Sigma_p$ which corresponds to the set of probability *mixtures* of S_p (i.e., the set of probability distributions over S_p).
- ***Def (loose)*:** A game of perfect information is a game in extensive form where the state is unambiguous.

MONTANA STATE UNIVERSITY

20

Mountains & Minds

Two Theorems

- ***Theorem* 1:** Every finite normal-form game has a mixed strategy equilibrium point (Nash).
  - This handles normal-form games.
- ***Theorem* 2:** Every finite game of perfect information has a pure strategy equilibrium point (Zermelo)
  - This handles extensive-form games.

MONTANA STATE UNIVERSITY

21

Mountains & Minds

Solving Matrix Games

- Recall the prisoner's dilemma

	Cooperate (Deny)	Defect (Confess)
Cooperate (Deny)	2/2	10/1
Defect (Confess)	1/10	**5/5**

Identifying the Equilibrium

- Pure strategy equilibrium
  - Consider mixed later
- Dominance
  - Dominance solvable
  - Only one dominant strategy
- Successive elimination of dominated strategies
- Cell-by-cell inspection

Dominant Strategy

- A strategy that outperforms all other choices no matter what opposing players do
- Player 1's strategies: { A, B, C }
- Player 2's strategies: { X, Y, Z }
- C is strictly dominant for Player 1 if:
  - $f(C,X) > f(A,X)$ $\quad$ $f(C,X) > f(B,X)$
  - $f(C,Y) > f(A,Y)$ $\quad$ $f(C,Y) > f(B,Y)$
  - $f(C,Z) > f(A,Z)$ $\quad$ $f(C,Z) > f(B,Z)$
- C is weakly dominant for Player 1 if:
  - Some inequalities are weak ($\geq$), at least one is strong($>$)

Dominance Solvable

Commandment

If you have a dominant strategy, use it.

Expect your opponent to use his or her dominant strategy if she has one.

- If each player has a dominant strategy, the game is *dominance solvable*

MONTANA STATE UNIVERSITY

25

Mountains & Minds

An Economic Game

- Two firms (magazine publishers) competing over sales
- *Time* and *The Economist* must decide upon the cover story to run some week.
- The big stories of the week are:
  - A presidential scandal (labeled S), and
  - A proposal to deploy US forces to Grenada (G)
- Neither knows which story the other magazine will choose to run

MONTANA STATE UNIVERSITY

26

Mountains & Minds

One Dominant Strategy

		The Economist	
		G	S
Time	S	100 , 100	0 , 90
	G	80 , 100	80 , 90

- Who has a dominant strategy?
- Assume it will be played!
- Other player can plan accordingly.

MONTANA STATE UNIVERSITY

27

Mountains & Minds

Dominated Strategies

		The Economist G	The Economist S
Time	S	100 , 100	0 , 90
	G	80 , 100	80 , 90

- Dominated Strategy:
  - There exists another strategy which always does better regardless of opponents' actions
- For *The Economist*: G dominant = S dominated

MONTANA STATE UNIVERSITY

Successive Elimination

- If a strategy is dominated, eliminate it.
- The size and complexity of the game is reduced.
- Eliminate any dominated strategies from the reduced game.
- Continue doing so successively.

MONTANA STATE UNIVERSITY

Example: Tourists and Natives

- Two bars (bar 1, bar 2) compete
- Can charge price of $2, $4, or $5
- 6000 tourists pick a bar randomly
- 4000 natives select the lowest price bar

		Bar 2 $2	Bar 2 $4	Bar 2 $5
Bar 1	$2	10 , 10	14 , 12	14 , 15
	$4	12 , 14	20 , 20	28 , 15
	$5	15 , 14	15 , 28	25 , 25

MONTANA STATE UNIVERSITY

Successive Elimination

- Does any player have a dominant strategy?
- Does any player have a dominated strategy?
  - Eliminate the dominated strategies
  - Reduce the normal-form game
  - Iterate the above procedure
- What is the equilibrium?

MONTANA STATE UNIVERSITY

Mountains & Minds

Successive Elimination

		Bar 2 $2	Bar 2 $4	Bar 2 $5
	~~$2~~	~~10 , 10~~	~~14 , 12~~	~~14 , 15~~
Bar 1	$4	~~12 , 14~~	20 , 20	28 , 15
	$5	~~15 , 14~~	15 , 28	25 , 25

		Bar 2 $4	Bar 2 $5
Bar 1	$4	20 , 20	~~28 , 15~~
	~~$5~~	~~15 , 28~~	~~25 , 25~~

MONTANA STATE UNIVERSITY

Mountains & Minds

No Dominated Strategies

- Often there are no dominated strategies.
  - Or: reducing the game is not sufficient.
- There may be multiple equilibria.
- Method:
  - Cell-by-cell inspection
- Ask:
  - Is each player playing the best response to the other player?

MONTANA STATE UNIVERSITY

Mountains & Minds

Solving Matrix Games

- Recall stone-paper-scissors

?

	Stone	Paper	Scissors
Stone	0	–1	1
Paper	1	0	–1
Scissors	–1	1	0

?

MONTANA STATE UNIVERSITY

34

Mountains & Minds

Stone-Paper-Scissors

- This game has no pure-strategy equilibrium point.
- The game does have a mixed-strategy equilibrium point, which may be apparent.
- Specifically, select any of the strategies (regardless of player) with probability 0.33.
- How do we find this?

MONTANA STATE UNIVERSITY

35

Mountains & Minds

Employee Monitoring

- Employees can work hard or shirk
  - Salary: $100K unless caught shirking
  - Cost of effort: $50K
- Managers can monitor or not
  - Value of employee output: $200K
  - Profit if employee doesn't work: $0
  - Cost of monitoring: $10K

MONTANA STATE UNIVERSITY

36

Mountains & Minds

Employee Monitoring

		Manager: Monitor	Manager: No Monitor
Employee	Work	50 , 90	50 , 100
	Shirk	0 , -10	100 , -100

- Best replies do not correspond
- No equilibrium *in pure strategies*
- What do the players do?

Finding Mixed Strategies

- Suppose:
  - Employee chooses (shirk, work) with probabilities (p,1-p)
  - Manager chooses (monitor, no monitor) with probabilities (q,1-q)
- Find expected payoffs for each player
- Use these to calculate best responses

Employee's Payoff

- First, find employee's expected payoff from each pure strategy
- If employee works: receives 50

$E_e(\text{work}) = 50 \times q + 50 \times (1 - q)$

$= 50$

- If employee shirks: receives 0 or 100

$E_e(\text{shirk}) = 0 \times q + 100 \times (1 - q)$

$= 100 - 100q$

Employee's Best Response

- Next, calculate the best strategy for possible strategies of the opponent
- For $q < 1/2$: SHIRK

  $E_e(\text{shirk}) = 100 - 100q > 50 = E_e(\text{work})$
- For $q > 1/2$: WORK

  $E_e(\text{shirk}) = 100 - 100q < 50 = E_e(\text{work})$
- For $q = 1/2$: INDIFFERENT

  $E_e(\text{shirk}) = 100 - 100q = 50 = E_e(\text{work})$

MONTANA STATE UNIVERSITY

40

Mountains & Minds

Manager's Best Response

- $E_m(\text{mntr}) = 90 \times (1 - p) - 10 \times p$
- $E_m(\text{no mntr}) = 100 \times (1 - p) - 100 \times p$
- For $p < 1/10$: NO MONITOR

  $E_m(\text{mntr}) = 90 - 100p < 100 - 200p = E_m(\text{no mntr})$
- For $p > 1/10$: MONITOR

  $E_m(\text{mntr}) = 90 - 100p > 100 - 200p = E_m(\text{no mntr})$
- For $p = 1/10$: INDIFFERENT

  $E_m(\text{mntr}) = 90 - 100p = 100 - 200p = E_m(\text{no mntr})$

MONTANA STATE UNIVERSITY

41

Mountains & Minds

Mixed Strategy Equilibrium

- Employees shirk with probability 1/10
- Managers monitor with probability 1/2
- Expected payoff to employee:

$$\tfrac{1}{10}[\tfrac{1}{2}0 + \tfrac{1}{2}100] + \tfrac{9}{10}[\tfrac{1}{2}50 + \tfrac{1}{2}50] = 50$$

- Expected payoff to manager:

$$\tfrac{1}{2}[\tfrac{9}{10}90 - \tfrac{1}{10}10] + \tfrac{1}{2}[\tfrac{9}{10}100 - \tfrac{1}{10}100] = 80$$

MONTANA STATE UNIVERSITY

42

Mountains & Minds

Properties of Equilibrium

- Both players are indifferent between any mixture over their strategies
- E.g. Consider employee strategies given mixed strategy for employer:
  - If shirk: $[\frac{1}{2}0+\frac{1}{2}100]=50$
  - If work: $[\frac{1}{2}50+\frac{1}{2}50]=50$
- Regardless of what employee does, expected payoff is the same

Indifference

	1/2 Monitor	1/2 No Monitor	
9/10 Work	50 , 90	50 , 100	= 50
1/10 Shirk	0 , -10	100 , -100	= 50
	= 80	= 80	

Linear Programming

Primal

Let f^{π} be the expected value of the game

minimize f^{π} subject to:

$$\sum_{s_1 \in S_1} p(s_1) = 1, \text{ where } p(s_1) \ge 0$$

$$\forall s_1 \in S_1, \sum_{s_2 \in S_2} p(s_1) f(s_1, s_2) - f^{\pi} \le 0$$

Dual

maximize f^{π} subject to:

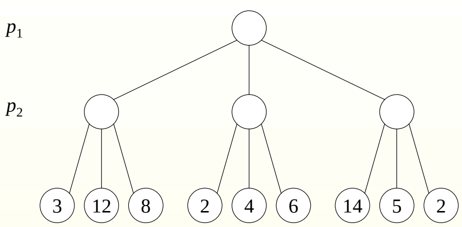
$$\sum_{s_2 \in S_2} p(s_2) = 1, \text{ where } p(s_2) \ge 0$$

$$\forall s_2 \in S_2, \sum_{s_1 \in S_1} p(s_2) f(s_1, s_2) - f^{\pi} \ge 0$$

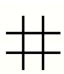
Solving Large Games

2001 A Space Odyssey

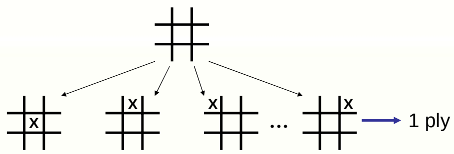
Game Trees



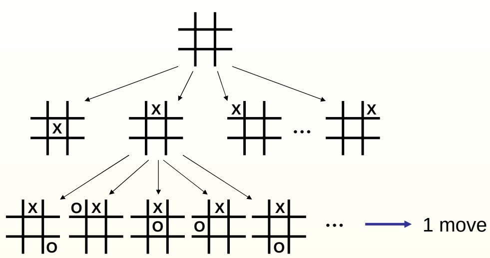
Game Trees



Game Trees



Game Trees



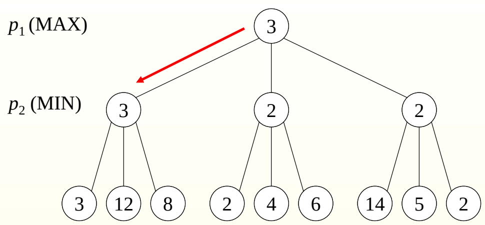
Solving Game Trees

- These are the most common types of games studied in artificial intelligence.
- In theory, you can expand the game tree until termination and then propagate the outcome up the tree.
- Approach used is the *minimax* algorithm (assuming zero-sum game).
- *Negamax* is similar except “payoff” is negated with each ply.

Minimax

- Create a start node as MAX node with current configuration.
- Expand tree
- Evaluate leaves
- Back up values, alternating between MIN and MAX until root is reached.
- Select move whose backed up value is value at root.

Minimax Example



Properties of Minimax

- Completeness: Minimax is complete if the tree is finite.
- Optimality: Minimax is optimal if playing against an optimal opponent.
- Time Complexity: $O(b^m)$
- Space Complexity: $O(bm)$, assuming depth-first exploration.
- Note, for chess, $b \approx 35$ and $m \approx 100$ for "reasonable" games.

Problems

- Interesting games have too many states to expand to the leaves.
- Number of nodes to expand is exponential in the depth of the tree and the branching factor.

MONTANA STATE UNIVERSITY

55

Mountains & Minds

Solutions

- Apply a heuristic evaluation function to estimate outcome without full expansion of the tree (like A*).
- Apply a cut-off procedures to prevent searching all of the branches of the tree (like A*).

MONTANA STATE UNIVERSITY

56

Mountains & Minds

Alpha-Beta Pruning

- Adapts minimax to consider the values of the nodes and whether exploration down a branch could possibly be fruitful.
- Proceeds from the following idea:
  - If you have an idea that is surely bad, don't take the time to see how truly awful it is (Pat Winston).

MONTANA STATE UNIVERSITY

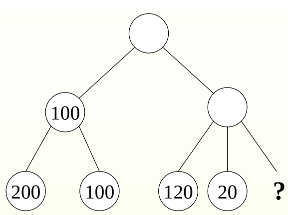
57

Mountains & Minds

A Simple Example

- Consider a situation where we are searching a game tree to depth 2 and the children are visited left-to-right.
- Suppose we have just visited the second child of “min” off of the second child of “max” and have computed the evaluation function to be 20.

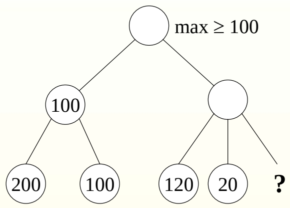
A Simple Example



A Simple Example

- Now we are considering whether or not to generate the third child.
- Note that the first “min” child of “max” has a value of 100.
- For “max” to select a move given by the second child, that child must have “min” value *greater than* 100.

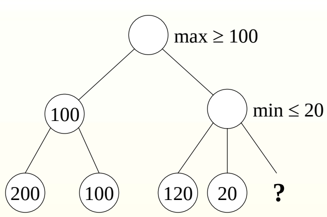
A Simple Example



A Simple Example

- But one of the children of “min” has been found to have value 20.
- We know that min will *not* select a node worse than (i.e., higher than) 20; therefore, further expanding this branch for “max” is futile.
- This line of reasoning sets up the *alpha-beta* procedure

A Simple Example



Alpha-Beta Procedure

- Traverse the game tree in depth-first, left-to-right order.
- Assuming we stop the search at ply *d*, then at each of the nodes at that depth, we apply the static evaluation function.
- This value will be propagated up the game tree as in minimax.

MONTANA STATE UNIVERSITY

64

Mountains & Minds

Alpha-Beta Procedure

- At each non-leaf node, store two values— *alpha* and *beta*.
- Let *alpha* be the best (i.e., maximum) value found so far at a "max" node.
- Let *beta* be the best (i.e., minimum) value found so far at a "min" node.
- Note *alpha* is monotonically non-decreasing and *beta* is monotonically non-increasing as you travel *up* the tree.

MONTANA STATE UNIVERSITY

65

Mountains & Minds

Alpha-Beta Procedure

- Initially assign *alpha* = $-\infty$ and *beta* = ∞ at the root.
- Given a node *n*, cut off the search below that node (i.e., generate no more children) if
  - *n* is a "max" node and *alpha*(*n*) $\geq$ *beta*(*i*) for some "min" ancestor *i* of *n*, or
  - *n* is a "min" node and *beta*(*n*) $\leq$ *alpha*(*j*) for some "max" ancestor *j* of *n*.

MONTANA STATE UNIVERSITY

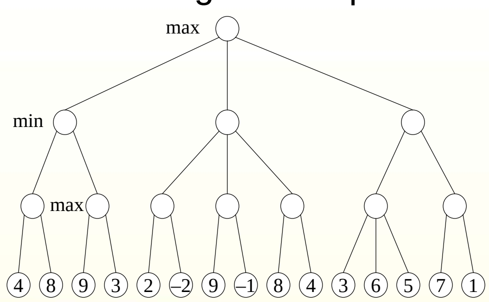
66

Mountains & Minds

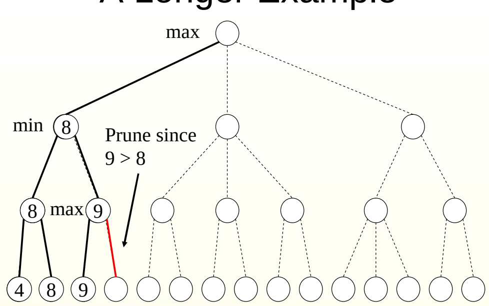
Alpha-Beta Procedure

- To avoid searching for the ancestor nodes to evaluate the above tests, we can carry *down* the tree the best values found so far at the ancestor.
  - At a "max" node *n*, *beta* is the minimum of the *beta* values at "min" node ancestors found so far.
  - At a "min" node *n*, *alpha* is the maximum of the *alpha* values at "max" node ancestors found so far.

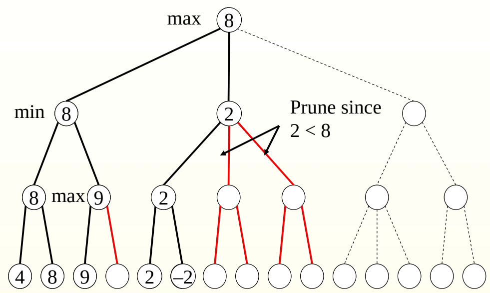
A Longer Example



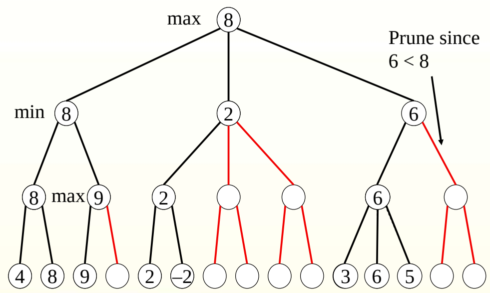
A Longer Example



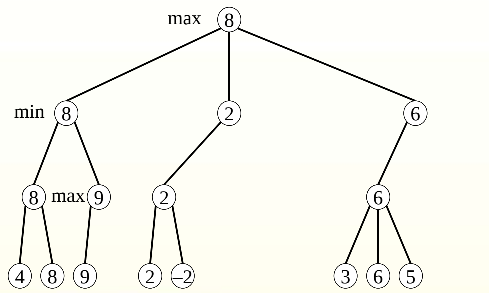
A Longer Example



A Longer Example



A Longer Example

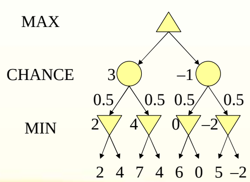


Alpha-Beta Summary

- *Alpha-beta* is guaranteed to compute the same minimax value for the root node as straight minimax (can you prove this?)
- Worst case—*alpha-beta* does no pruning and examines $O(b^m)$ leaf nodes
- Best case—*alpha-beta* examines only $O(2b^{m/2})$ leaf nodes thus enabling twice the depth to be searched.

Nondeterministic Games

- Games such as backgammon use dice (the chance player) to determine legal moves.
- Here is a simplified example with coin-flipping instead of dice-rolling.



Solving Nondeterministic Games

- Expectimax gives perfect play.

$$Expectimax(C) = \sum_i P(d_i) \max_{s \in S(C,d_i)} (utility(s))$$

- It is just like Minimax, except we must also handle chance nodes.

…
if *state* is a chance node **then**
  return average of Expectimax-Value of Succ(*state*)
…

- A version of α-β pruning is also possible, but only if the leave nodes are bounded.

Complexity of Expectimax

- If the player knew in advance all the dice rolls that would occur, the game would reduce to minimax.
- Since Expectimax also considers all possible dice rolls, it will take $O(b^m n^m)$, where n is the number of distinct rolls.
- Thus, dice rolls effectively increase the branching factor: 21 possible unique rolls with 2 dice.
- Consider backgammon with about 20 legal moves:
  - depth 4 = $20 \times (21 \text{ x } 20)^3 \approx 1.2 \times 10^9$
- As depth increases, probability of reaching a given node shrinks accordingly. Thus the value of look-ahead is diminished.

State of the Art in Games

- Chess
  - Received largest share of attention in computer game research.
  - Deep Blue defeated Gary Kasparov in 6-game match in 1997.
  - Deep Blue searches 200 million positions per second.
  - Deep blue has a highly sophisticated evaluation function.
  - At times, look-ahead reaches 40-ply.

State of the Art in Games

- Checkers
  - In 1952, Arthur Samuel created learning program to play checkers—achieved reasonably good play.
  - Chinook uses α-β and a large database for 8-piece positions with perfect play.
  - Chinook won 1992 US Open and was declared world champion in 1994 when champion, Marion Tinsley, withdrew for health reasons.
  - Recent announcement that Chinook database of moves is now “unbeatable.”

State of the Art in Games

- Othello
  - Widely studied as a computer game.
  - Typically, computer-based Othello players are much better than humans, due to the relatively small search space.
  - Human champions refuse to play against computers, because the computers are too good.

MONTANA STATE UNIVERSITY

79

Mountains & Minds

State of the Art in Games

- Backgammon
  - Interesting game to study due to “chance” element and “gambling” element.
  - In 1990, BKG beat the world-champion 5-1. It is believed it was due to the luck of the dice.
  - In 1992, Gerry Tesauro at IBM applied temporal difference learning with a neural network to produce TD-Gammon.
  - TD-Gammon has beaten or been competitive with current world-champions, even with less favorable dice rolls.

MONTANA STATE UNIVERSITY

80

Mountains & Minds

State of the Art in Games

- Go
  - Branching factor is over 300.
  - Most programs are simply awful.
  - As a result, human champions refuse to play against computer opponents.
  - Due to large branching factor, plied search tends to be limited, instead favoring pattern matching in large, patterned databases.
  - Recent announcement that a computer (Huygens) has now beaten a Go professional (2008).
  - Applied Monte Carlo Tree Search.

MONTANA STATE UNIVERSITY

81

Mountains & Minds

State of the Art in Games

- Differential Games
  - Models games played in continuous state spaces with continuous actions.
  - Dynamics modeled with differential equations.
  - Recent as a research topic in artificial intelligence.
  - Most computer-based games make simplifying assumptions, such as pure strategies, quantized state space, quantized action space, and Markov assumption.

MONTANA STATE UNIVERSITY

82

Mountains & Minds