Lecture 9:Multi-Player Games

AI For Traditional Games
Prof. Nathan Sturtevant
Winter 2013

UNIVERSITY OF DENVER

UNIVERSITY OF DENVER

Pruning properties

- Techniques covered so far only prune if the sum of 2 players scores adds up to maxsum
- Consider Hearts as a maximization game:
  - Score is points taking by other players
  - maxp = 26; maxsum = 52; minp = 0; minsum = 52
  - The only way 2 players scores sum to 52 is if they both get 26; but then we just do immediate pruning!
  - Shallow pruning never occurs!
  - Depends on properties of game, not order of tree

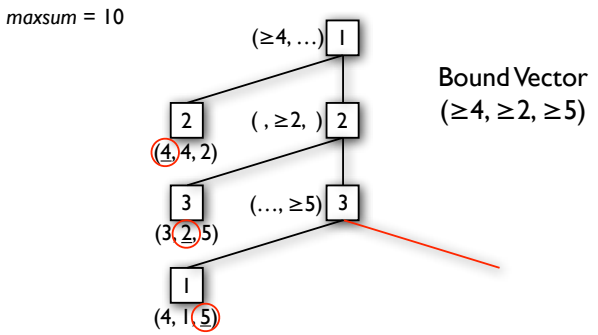
UNIVERSITY OF DENVER

Multi-player games have more than 2 players

- All pruning that we've seen so far only involves bounds for 2 players
  - It is highly unlikely that 2 players have their bounds add to maxsum
  - Effective pruning techniques involve bounds from all players

UNIVERSITY OF DENVER

Last branch pruning (full)



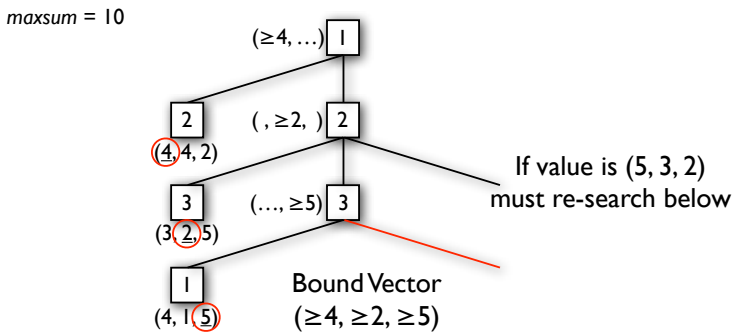
UNIVERSITY OF DENVER

Speculative Pruning

- When *b* gets large, the gains of speculative pruning will be minimal; can we do better
- If we analyze where deep pruning failed, (depth 3 tree) there were two conditions:
  - Unseen leaf at P2 node has better score for both P2 and P1
  - Unseen leaf at P3 node is better for P3 and P2
- This corresponds to having the move ordering wrong
- Solution: prune, and then verify these two conditions

UNIVERSITY OF DENVER

Speculative Pruning Example



UNIVERSITY OF DENVER

Speculative Pruning Analysis

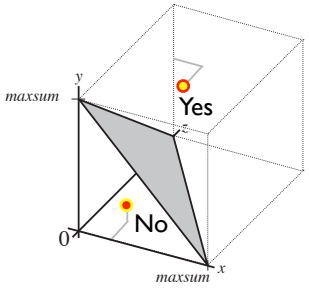
- Speculative pruning depends on the order of nodes in the tree, not the values in the game
- Asymptotically still must provide a strategy for each player, so bounded by $b^{(n-1/n)d}$
  - Can verify this lower-bound via recurrence equations

UNIVERSITY OF DENVER

Generalized Pruning

- In practice our evaluation functions are points in 3d space
  - Each bound drops a plane in 3d space
  - If a point is guaranteed to be outside the current 3d bounded space, we have the potential to prune
- Separates the pruning decision from the algorithm that implements correct pruning

Decision view



Generalized Pruning

- Suppose that the evaluation function is a function of the underlying outcomes of the game
  - eg outcomes far fewer than strategies for play
  - Then, we can iterate over outcomes to see if pruning is possible

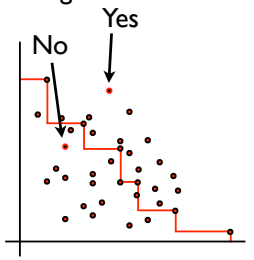
Example of complex non-zero sum game

Possible Tricks		Game Score		Evaluation Fcn
(0, 0, 3)		(-20, …)		(0, …)
(0, 1, 2)		(-20, …)		(0, …)
(0, 2, 1)		(-20, …)		(0, …)
(0, 3, 0)		(-20, …)		(0, …)
(1, 0, 2)	Player 1 bids 2	(-20, …)	⇨	(0, …)
(1, 1, 1)		(-20, …)		(0, …)
(1, 2, 0)		(-20, …)		(0, …)
(2, 0, 1)		(20, …)		(2, …)
(2, 1, 0)		(20, …)		(2, …)
(3, 0, 0)		(21, …)		(1, …)

Generalized technique

Generate points for the utility of each game outcome.

0. Calculate current bound vector and heuristics.
1. Check hash table for entry.
2. Remove values that are inconsistent with heuristic.
3. Check boundary of space.
4. Add result to hash table.



UNIVERSITY OF DENVER

Other techniques?

- History Heuristic / move ordering
  - No problem...but…may change equilibrium strategy/ payoff
- Transposition tables
  - No problem...but…may change equilibrium strategy/ payoff

UNIVERSITY OF DENVER

Alternate: Paranoid

- Two-player games give us better properties than multi-player games
  - Reduce a game to two players by assuming the other players formed a coalition against “us”
  - Actually just a revised evaluation function
    - Given the same eval, max^n would play the same
- Is the tradeoff worthwhile?
  - Restricted evaluation function vs deeper search

UNIVERSITY OF DENVER

Paranoid Analysis

- From a pruning perspective, same analysis as alpha-beta pruning
  - But, the joint player has a branching factor of $b^{n-1/n}$
- Paranoid makes the assumption that a player will take an arbitrarily large loss in score in return for a arbitrarily small penalty to the paranoid player
  - If the game doesn’t facilitate this (eg Chinese Checkers), then it is find to use
  - Doesn’t work nearly as well in card games

UNIVERSITY OF DENVER

Alternate: Best-Reply Search

- Abstraction of full games
  - Only allow a single opponent to perform actions
  - Opponent has branching factor $b \cdot n$
- Results in much smaller game tree
  - Game states during search may be illegal
- Is the tradeoff worthwhile?
  - Restricted game model vs deeper search

UNIVERSITY OF DENVER

Opponent modeling

- What if I don't have the right opponent model?
  - Tie-breaking (shared eval function)
    - Break ties against player at root of tree
    - Will work poorly if there are lots of ties
  - Tie-breaking (bad evaluation function)
    - Need to compare values with epsilon difference
    - Or, simplify opponents' model to represent uncertainty of their actions

	Players A v. B	Player A		Player B	
		Score	%Win	Score	%Win
(a)	mOT_{MT} v. MT_{mOT}	248.6	74.7	163.8	25.3
(b)	mOT_{MT} v. MT_{MT}	235.4	59.0	199.2	41.0
(c)	mOT_{mOT} v. MT_{mOT}	198.2	53.5	191.4	46.5
(d)	mOT_{mOT} v. MT_{MT}	178.2	44.0	207.3	56.0
(e)	mOT_{MT} v. mOT_{mOT}	212.4	37.2	250.8	62.8
(f)	MT_{mOT} v. MT_{MT}	157.3	36.0	218.4	64.0

	Players A v. B	Player A				Player B	
		Score	%Win	%Gain	%Loss	Score	%Win
(a)	mOT_g v. MT_{mOT}	241.7	68.5	15.0	6.8	178.2	31.5
(b)	mOT_g v. MT_{MT}	218.2	53.5	9.5	5.5	206.1	46.5
(c)	mOT_g v. mOT_{MT}	242.2	54.8	4.8	8.0	228.7	45.2
(d)	mOT_g v. mOT_{mOT}	230.6	46.0	8.8	4.0	243.8	54.0