Lecture 7: Multi-Player Games

AI For Traditional Games
Prof. Nathan Sturtevant
Winter 2011

Review: Multi-player game tree

- All nodes are max nodes
- Scores/payoffs are $n$-tuples, with a value for each player
- Assume that players strictly want to increase their own payoff and don’t care about minimizing/maximizing other players payoff

Zero-sum / constant sum

- Have been looking at two-player zero-sum games
- Multi-player equivalent is constant-sum
- eg (3, 4, 2) (sum = 9)
  - Subtract 3 from each player: (0, 1, -1) (zero sum)

Max

Max$^n$(GameState *g, Player *p, depth)
if (game over or depth == 0) return p->eval(g)
best ← nil
for each successor s in 1… # successors
  ApplyMove(s)
  tmp = Max$^n$(g, p, depth-1)
  UndoMove(s)
  if (tmp[g->currPlayer()] > best[g->currPlayer()])
    best = tmp;
return best
Sample Tree

Max\textsuperscript{n} in equilibrium?

- Max\textsuperscript{n} generalizes minimax for \( n \) players
- All players are making locally maximal choices
  - If any player changes their strategy
    - They must get a lower score
    - Thus, and equilibrium
  - Note that the strategy change is local in the tree
    - Other players cannot adjust their strategies accordingly higher in the tree; those moves already made

Max\textsuperscript{n} tie-breaking & implications

- Each different tie-breaking strategy leads to a different strategy
- Blue / yellow strategies are separate equilibria
  - Cannot be mixed to produce an equilibrium strategy

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”
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 an 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

Shallow pruning

• First analyzed by (Korf, 91)

Can we prune in $\max^n$?

• With no restrictions on players score, no
• What restrictions do we have?
  • $\maxsum$, $\minsum$, $\maxp$, $\minp$
  • Assume $\maxsum = \minsum$ (constant sum)
  • Assume $\minp = 0$

Deep pruning

$maxsum = 10$

Bound vector:
$(5, 6, -)$
$5+6 \geq \maxsum$

$maxsum = 10$

$(7, 3, 0)$
or $(6, 2, 2)$

$(7, 3, 0)$

$(4, 1, 5)$
or $(0, 4, 6)$
Deep Pruning

- Deep pruning can fail in general
  - Bounds at the bottom of the tree cannot be the max\(^n\) value of the tree
  - But, propagation of effects can travel arbitrarily far up the tree
  - Zipper effect -- order of teeth at the bottom determines which one comes out on top

Last Branch Pruning (first attempt)

- Deep pruning fails because there are additional moves which can influence the values in the tree
- What if there were no other moves left?
  - No moves to influence propagation

Last Branch Pruning (First attempt)

\[ \text{maxsum} = 10 \]

2\text{nd player:}
1) on last branch
2) existing score worse for P1

Bounds crossed; we can prune