Robust Algorithms For Game Play Against Unknown Opponents

Nathan Sturtevant
University of Alberta
May 11, 2006

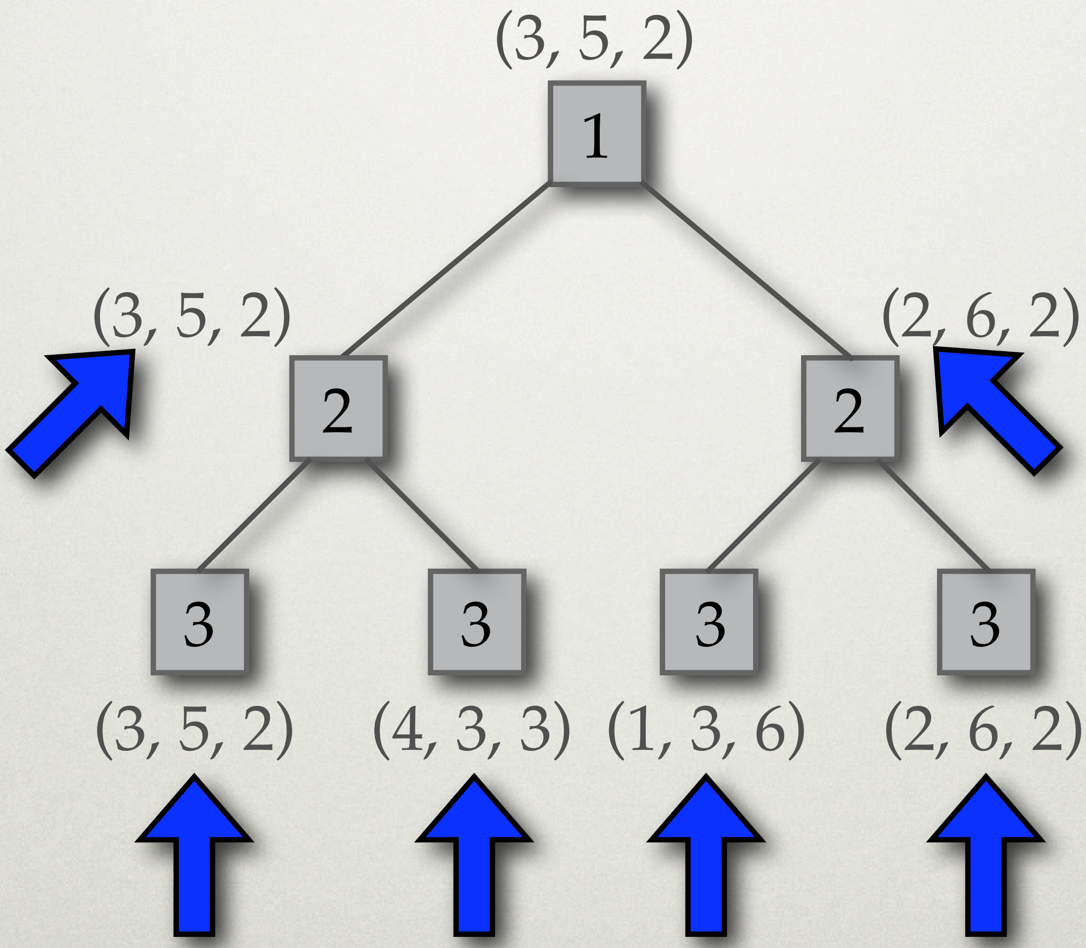
Introduction

- A lot of work has gone into two-player zero-sum games
- What happens in non-zero sum games and multi-player games?
    - Actual games
    - Robotic teams
- Perfect-information extensive-form

Multi-Player Games

- Max^n algorithm
  - Luckhardt and Irani, 1986
- *n*-tuple of scores/utilities
- One value for each player, eg (3, 5, 7)

Max^{N} Decision Rule



Max^N Computation

- Max^n computes an equilibrium strategy
  - If all players were given the strategy, nobody would have incentive to change
- Assumes:
  - All utilities known exactly
  - Tree analyzed completely
  - Players choose common strategy
  - Strategies cannot be changed

Sample Domain: Spades

- Spades
  - Trick-based card game
  - Use 3-player variation
  - Many similar card games
- Tricks → Hands → Game

Spades Rules - 1 Hand

- Cards dealt to players
- Players *bid* how many *tricks* they will take
- After playing the *hand*:
  - -10x*bid* if bid is missed (eg bid 5 take 4)
  - 10x*bid* if bid is made (eg bid 5 take 5 or 6)
  - -100 for taking 10 *overtricks*

Spades Strategies

- Players may play with different strategies:
  - Minimize overtricks (mOT)
  - Maximize tricks (MT)
- Players must model opponents' strategies

Experimental Setup

- 100 games, played to 300 points
- 7 cards per player
- Perfect information

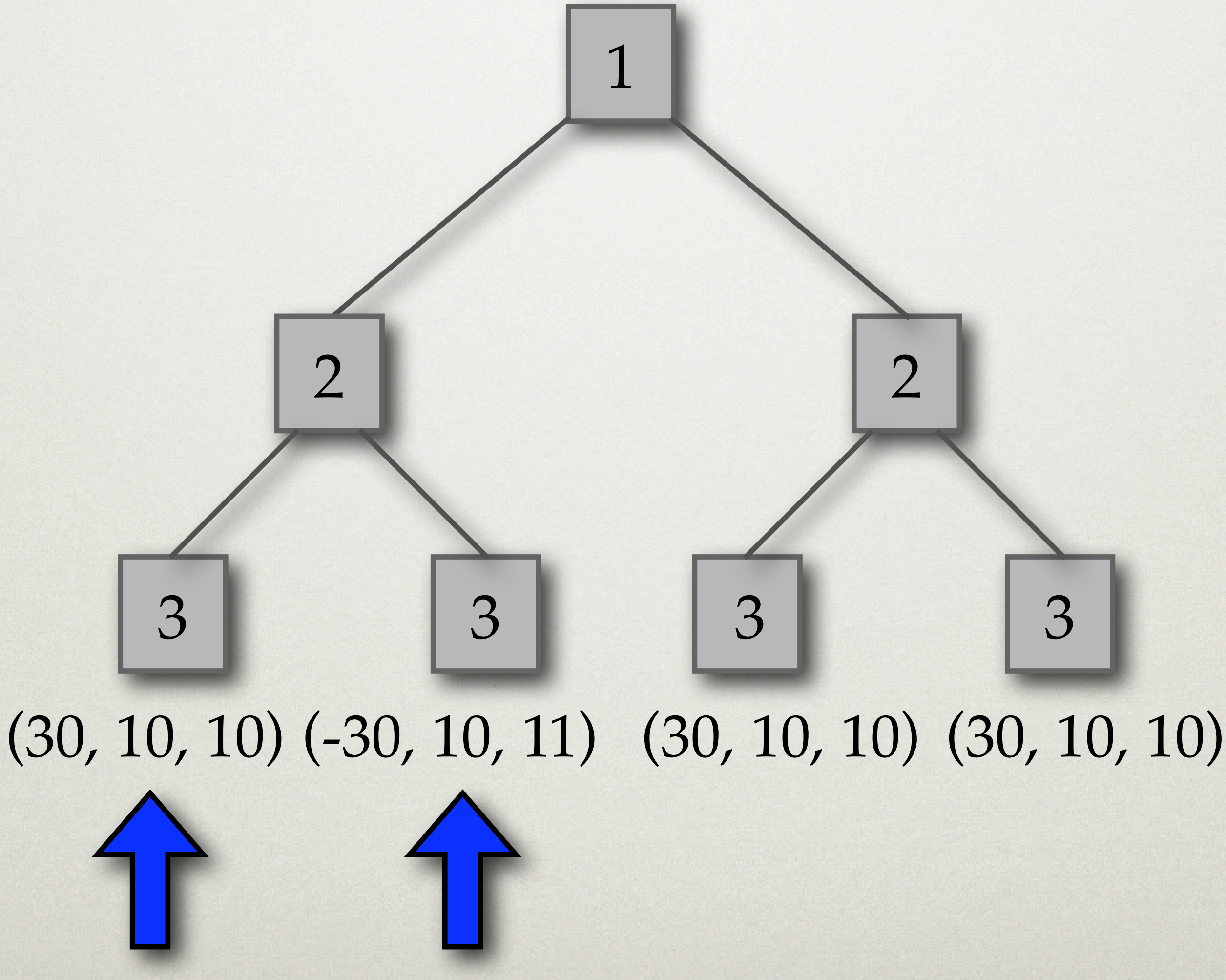
Experimental Results

		Player A		B
A	B	Score	%Win	Score
mOT	MT	178.2	44.0	207.3
mOT	MT	198.2	53.5	191.4
mOT	MT	235.4	59.0	199.2
mOT	MT	248.6	74.7	163.8

Results - Discussion

- We must use *some* opponent model
  - Don't know opponents utilities
    - Even in perfect-information games
    - Payoffs ≠ utilities
  - Model has large effect on quality of play

Spades Example



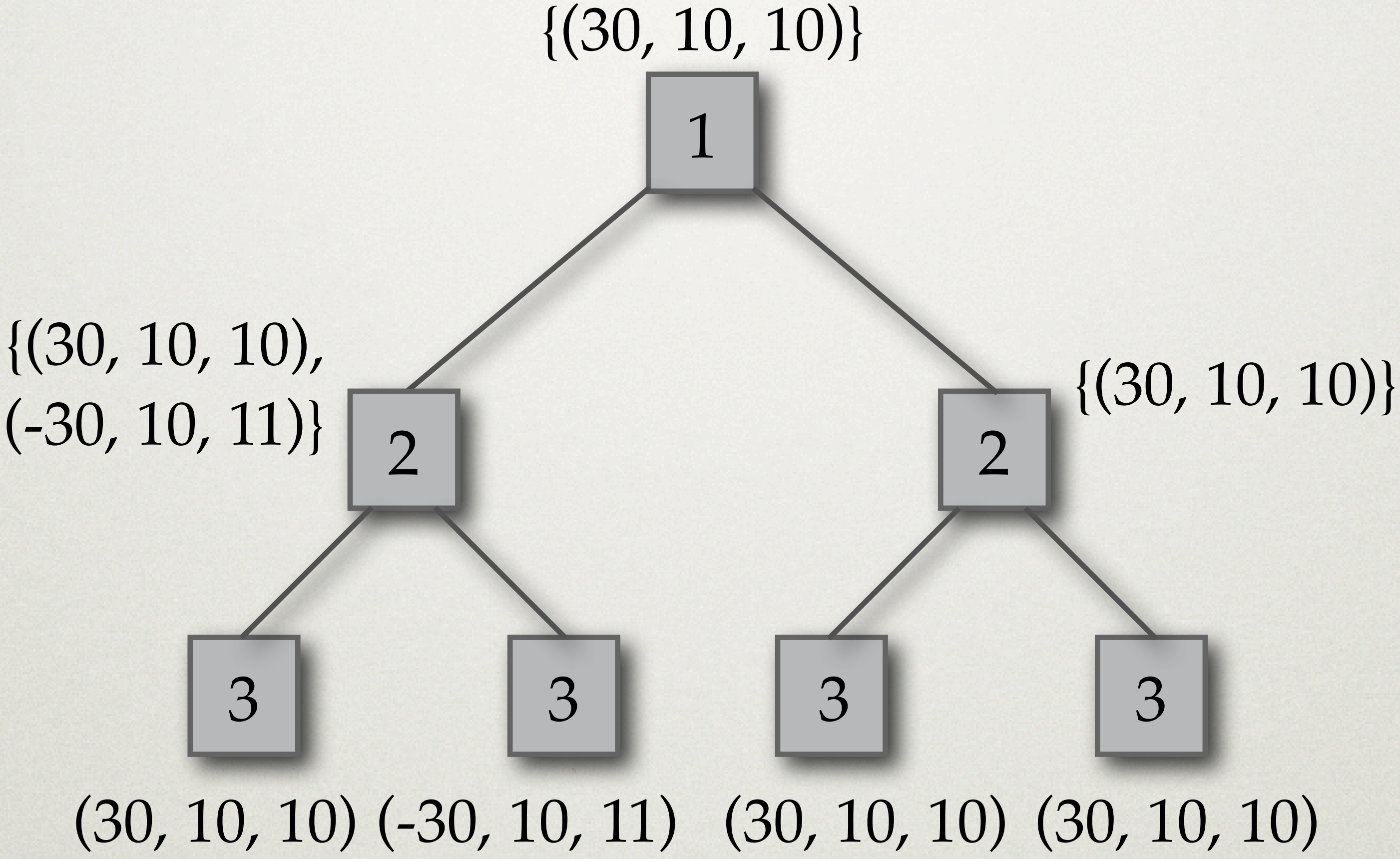
Max^n Deficiencies

- Max^n only calculates one of many equilibria
- Keeps no information about alternates
  - Some alternates may be less risky in the face of uncertain opponents

Soft-Max^n

- Back up sets of max^n values
  - Each time there is a tie, return both values
  - Calculates a superset of all equilibria

Spades Example



Soft-Max^n - Dominance

- Dominance relationship to compare max^n sets with respect to a given player
  - {(10, 2, 7), (8, 7, 4)} vs:
    - {(5, 10, 4)} – strictly dominates
    - {(8, 4, 7)} – weakly dominates
    - {(9, 1, 9)} – no domination
- Union all sets that are not dominated

Soft-Max^n - Outcomes

- How large can soft-max^n sets grow?
  - In trick-based card games
    - n players, c cards
    - $O(c^{n-1})$ possible game outcomes
  - In other domains we can reduce number of outcomes

Opponent Modeling

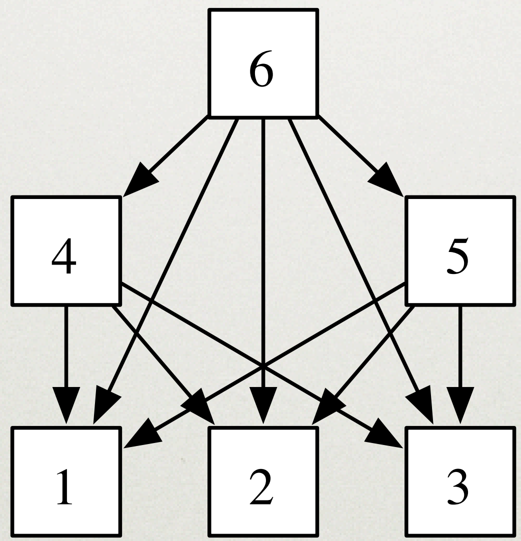
- Represent opponent models as a graph
  - Nodes are outcomes in the game
  - Directed edges represent preferences
  - Partial order over game outcomes

Opponent Models

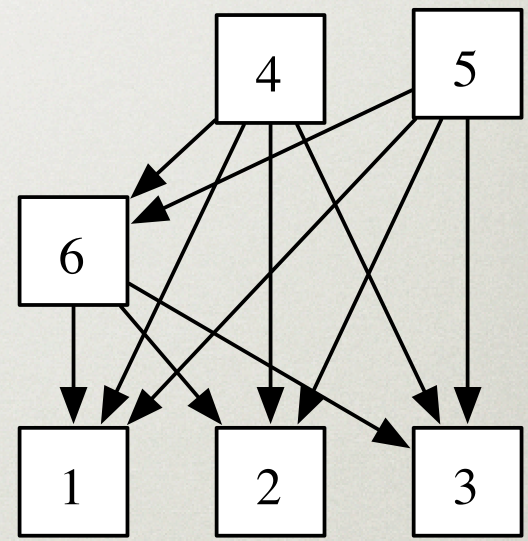
Possible Outcomes

1: (0, 0, 2)
2: (0, 1, 1)
3: (0, 2, 0)
4: (1, 0, 1)
5: (1, 1, 0)
6: (2, 0, 0)

maximize tricks



minimize overtricks



Opponent Modeling

- We do not want to assume too much about our opponents
  - Eliminating all ties would remove all ambiguities from max^n analysis
  - Analysis will be incorrect unless we have a perfect opponent model
  - More or less accurate model?

Opponent Models

- Combine opponent models to form more generic opponent models
  - Intersection of edges over each opponent model
  - Builds a generic opponent model

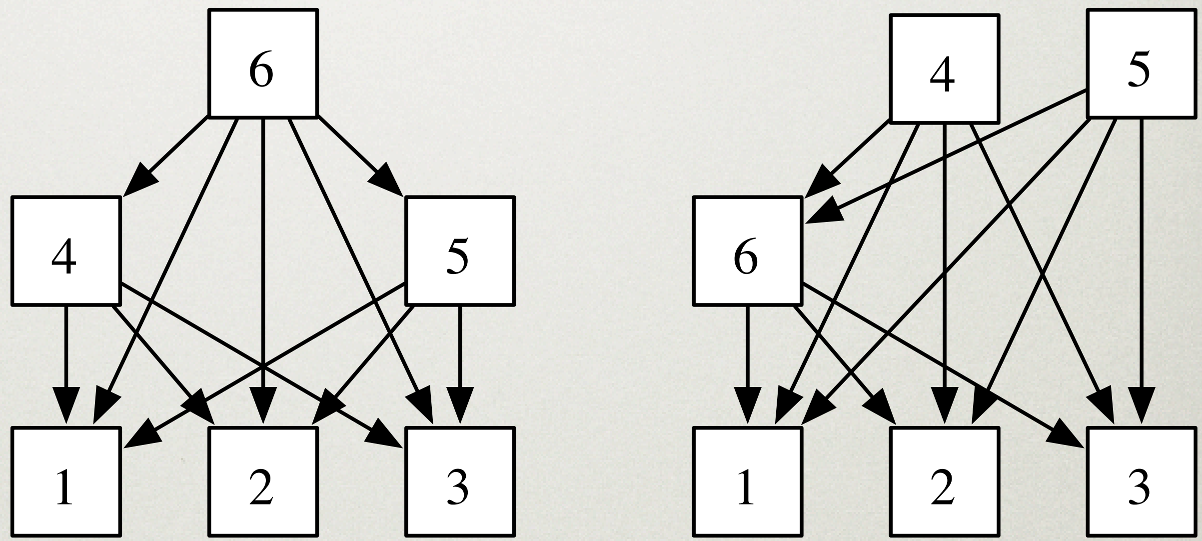
Opponent Models

Possible Outcomes

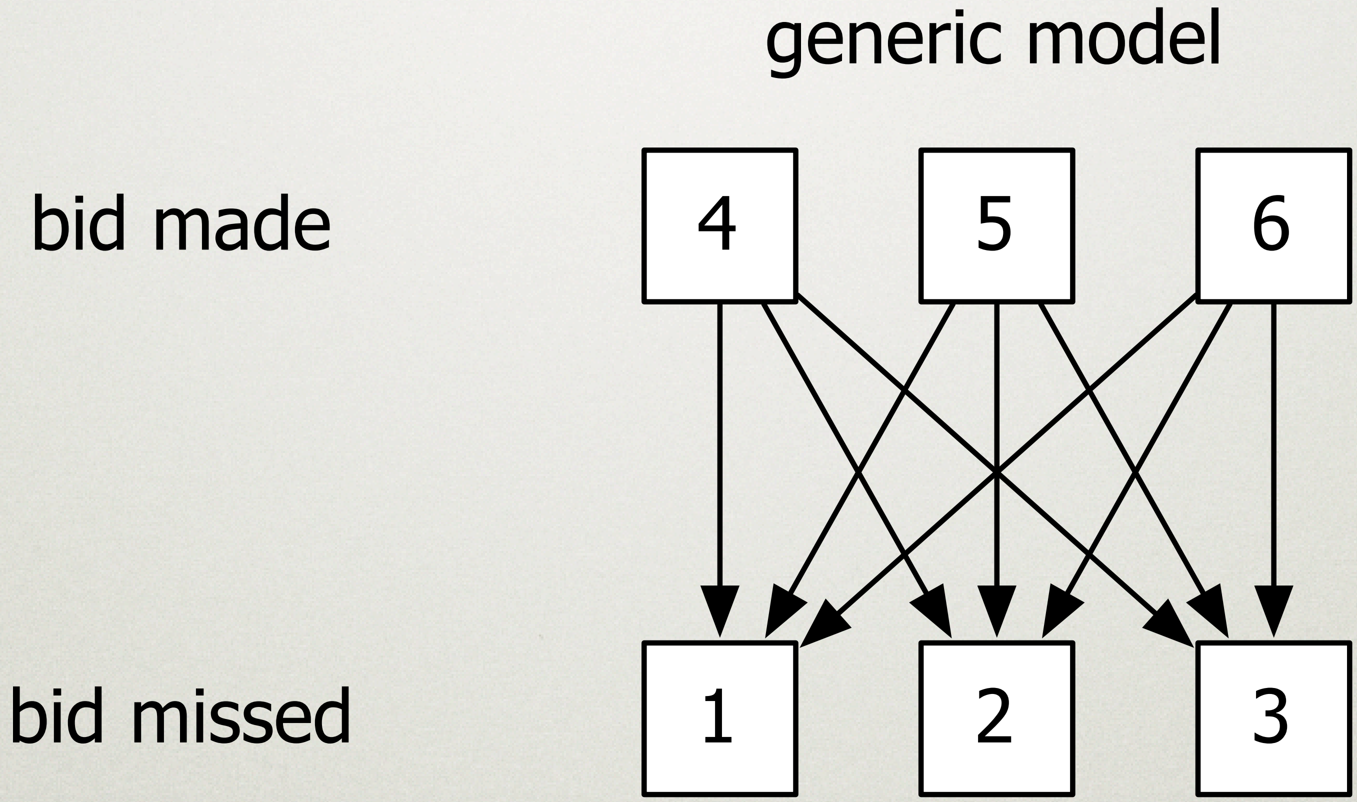
1: (0, 0, 2)
2: (0, 1, 1)
3: (0, 2, 0)
4: (1, 0, 1)
5: (1, 1, 0)
6: (2, 0, 0)

maximize tricks

minimize overtricks



Generic Opponent Model



Soft-Max^n Performance

- Run same experiments as before
  - Use soft-max^n with generic opponent models

Experimental Results

		Player A			
A	B	Score	%Win	%Gain	%Loss
mOT	MT	241.7	68.6	15.0	6.8
mOT	MT	218.2	53.5	9.5	5.5
mOT	mOT	242.2	54.8	4.8	8.0
mOT	mOT	230.6	46.0	8.8	4.0

Learning in Soft-Max^N

- We observe players' actions during the game
  - Sometimes we can distinguish between models based on their moves
  - Similar to version space learning
- Used 3 player models and did inference
  - In 900 hands, 423 (correct) inferences
  - Identify player type in 1/6 hands

Soft-Max^N Summary

- It is better to under-assume than over-assume about our opponents
- Need a bigger picture of what is happening in the game
- Can observe players to learn their models
- Only use a partial ordering of outcomes
  - No utilities actually used

Thanks

- Joint work with Michael Bowling
- See also:
  - ProbMax^n : Opponent Modeling in N-Player Games, Nathan Sturtevant, Michael Bowling, and Martin Zinkevich, AAAI-06.

University of Alberta
Quaecumque Vera
Informatics iCORE Circle of Research Excellence
Alberta Ingenuity Centre for Machine Learning