**1/29/18**

10-Expectimax-Search

When an agent takes an action, the outcome isn’t necessarily deterministic

* what then? - previously all actions are deterministic
* previous belief is that opponent will act similar to how we do
  + but not all players are like that - try to lose, random moves, etc.

**Uncertain Outcomes**

* unintentional randomness and intentional randomness
* Idea: Uncertain outcomes controlled by chance, not an adversary!
* Values should now reflect average-case (expectimax) outcomes, not worst-case (minimax) outcomes

**Expectimax search: compute the average score under optimal play**

* Max nodes as in minimax search
* Chance nodes are like min nodes but the outcome is uncertain
* Calculate their expected utilities ▪ I.e. take weighted average (expectation) of children

**Why wouldn’t we know what the result of an action will be?**

* Explicit randomness: rolling dice
* Unpredictable opponents: the ghosts respond randomly
* Actions can fail: when moving a robot, wheels might slip

Markov Decision Processes: formalize underlying uncertain-result problems

**expectimax pseudocode: (5/6)**

<https://courses.cs.washington.edu/courses/cse415/18wi/uwnetid/slides/10-Expectimax-Search.key.pdf>

No pruning for expectimax - requires knowledge of bounds etc.

**Depth-Limited Expectimax**

* limit depth - use estimate of true expectimax value (requires more work to compute real value)
  + exponential increase as depth of tree increases.

A **random variable** represents an event whose outcome is unknown

A **probability distribution** is an assignment of weights to outcomes

**In expectimax search, we have a probabilistic model of how the opponent (or environment) will behave in any state**

* Model could be a simple uniform distribution (roll a die)
* Model could be sophisticated and require a great deal of computation
* We have a chance node for any outcome out of our control: opponent or environment
* The model might say that adversarial actions are likely!

**Multi-Agent Utilities**

Generalization of minimax:

* Terminals have utility tuples
* Node values are also utility tuples
* Each player maximizes its own component
* Can give rise to cooperation and competition dynamically…
* Utilities: