Expectiminimax Algorithm

- **EXPECTIMINIMAX** gives perfect play for non-deterministic games.

- Like MINIMAX, except add chance nodes:
  - For **max** node return highest **EXPECTIMINIMAX** of SUCCESSORS.
  - For **min** node return lowest **EXPECTIMINIMAX** of SUCCESSORS.
  - For **chance** node return **average** of **EXPECTIMINIMAX** of SUCCESSORS.

- Here exact values of evaluation function do matter (“probabilities”, “expected gain”, not just order).

- \(\alpha-\beta\) pruning possible by taking weighted averages according to probabilities.
*-Minimax

- $\alpha$-$\beta$ pruning possible by taking weighted averages according to probabilities

- *-Minimax (B. Ballard, 1983)
  - 50% improvement with random node order
  - Order of magnitude improvement with optimal order

- Add *cut-offs to chance nodes*
  - Max and min nodes as in alpha-beta algorithm
  - Assume that all branches not searched have the worst-case result
  - Assume range of evaluating values is *bound by interval* $[L, U]$
**-Minimax Cut-Off

**Alpha cut-off** in chance node with $N$ equally likely children

$$\frac{1}{N} \left( \sum_{i=1}^{l-1} V_i + V_l \right) + \frac{V_l}{N} + U \ast (n - l) \leq \alpha$$

**Beta cut-off** in chance node with $N$ equally likely children

$$\frac{1}{N} \left( \sum_{i=1}^{l-1} V_i + V_l \right) + \frac{V_l}{N} + L \ast (n - l) \geq \beta$$

Bounds passed to $C$ from $A$ with $[L, U] = [-10, 10]$

$$- \frac{1}{3} (2 + V_l + L \ast 1) \geq \beta \Rightarrow V_l \geq 20$$

$$- \frac{1}{3} (2 + V_l + U \ast 1) \leq \alpha \Rightarrow V_l \geq -3$$