Games with Randomness

(TOO BRIEF)

6.1 Showcase Showdown in The Price is Right

Consider the spin in Price is Right. Three players in turn try to come closest to 100 without going over. They spin a wheel with an equal chance of 5,10,15,…,95,100. Their only decision is whether to keep the result of the first spin ("freeze"), or to spin a second time and their total is the sum of the two spins. A person going over 100 is out. If there is a tie, the earlier player is the winner.

We can work back from the last spinner. If their first spin exceeds the best so far, they win. If not, they spin again. Based on that, we can infer the probability that they would win, given the score at that point. For the second player, they too have to spin again if their first spin is insufficient. But if it is sufficient, they have a decision. They should pick the move that maximizes the probability of victory.

Some mathematical notation. Let $b(s)$ be the probability of the third player winning, given the score is s at the start of their go. Then we can calculate $b(s)$ with a formula:

The idea is that they win immediately if their first spin is higher. This has chance $1 - s/100$. Failing which, they win if their second spin takes them higher but not over 100. The chance of a correct second spin is again $1 - s/100$. Adding the two together we get:

$$b(s) = (1 - s/100) + s/100 \times (1 - s/100) = 1 - s^2/10000.$$

Then we let $a(s)$ be the probability of the second player winning, given the score is s at the start of their go. Once we calculate $b(s)$ for all s, we can calculate $a(s)$ for all s, using a spreadsheet, for example. This yields

> THEOREM: *First players sticks at 70 or higher, but spins a second time at less than that. Second player sticks at 60 or higher if greater than first player, otherwise spins a second time.*

6.2 Bearing Off

Recall BearingOff. Here the randomness occurs first, and the player then chooses the move. Again for a given configuration of counters for the player and her opponent, we pick the move that maximizes the chance of success. It's very like MiniMax on the tree, except that there is averaging as well as maximizing happening simultaneously.

One way to set it out is to think of nodes in the game tree where the Randomness occurs. In the extension to Minimax, these nodes calculate their value by the average/expected value of their children.

Notes:

- it is important that if one uses a static analyzer, that the function closely approximate the chance of winning (and is not just a ranking function).
- There is an extension of α–β pruning.
- If we calculate things exactly, one might use the algorithmic technique dynamic programming instead, but its equivalent.