Optimal Strategy for The Weakest Link

Aaron Kruchten

Introduction

This write up contains the portion that I contributed in the final project for 21-393 Operations Research II at Carnegie Mellon University in the fall of 2019.

For this project, we attempt to derive an optimal player strategy for the game show, *The Weakest Link*. The rule of the game show are as follows. There are N players, typically 9, that play (N-1) 'standard' rounds where the contestants form a chain along which they will answer various trivia questions. Questions are asked through the chain for increasing amounts of money until the chain is broken when a contestant either answers a question incorrectly or they 'bank' the current money earned. At the end of each of these rounds, all the players in the chain vote remove one player, 'the weakest link.'

Once the rounds have been played and 2 players remain, they play a head-to-head format under which they each answer 5 questions - the person to answer the most questions correct then wins. In the event of a tie, the game goes into 'sudden death' where the players are asked questions until a player answers a question incorrectly and the next player answers their question correctly. The player that remains wins the entire pot earned. All other players earn no money.

The goal of any player is to maximize their expected earnings, and since the game has multiple stages, the best approach is to work backwards. We first solve the final (head-to-head) round analytically. To solve the standard rounds, the optimal banking strategy must first be solved; this is done through dynamic programming. Since the optimal banking strategy does not leave us with a closed-form solution, the voting strategy for the penultimate round is approached empirically. (This final empirical solution is not included in this writeup because that was done by other members of my group.)

Final Round Probability Calculation

In this section, we will compute the probability that a player wins in the final head-to-head round of *The Weakest Link*. We let player one and player

two be the two players competing in the final round with probability p_1 and p_2 respectively that they will answer any given question correct. We will compute the probability that player one wins.
The problem can be separated into two sub-problems. The probability that player one wins during the five questions, and the probability that player one wins in the sudden death round.

Five Round Problem
Player one wins in this round if they answer more questions correct than player two. The number of questions each player answers correctly follows a binomial probability mass function, and therefore the probability that player one wins in this round is the following.

$\sum_{k=1}^{5} \binom{5}{k} p_1^k (1-p_1)^{5-k} \sum_{l=0}^{k-1} \binom{k-1}{l} p_2^l (1-p_2)^{5-l}$

Probability They Enter Sudden Death
The players enter sudden death if they both answer the same number of questions correctly. This is the product of two binomial probability mass functions with an equal number of successes and successes ranging from 0 to 5.

$\sum_{k=0}^{5} \binom{5}{k}^2 (p_1p_2)^k ((1-p_1)(1-p_2))^{5-k}$

Probability P_1 Wins Sudden Death
Consider the case where player one wins on the i_{th} round with $i \in \mathbb{N} \setminus \{0\}$. This could only happen if the players either answer all the previous $i-1$ questions both correct or both incorrect. The probability of both players answering a question correct or incorrect is $p_1p_2 + (1-p_1)(1-p_2)$. Therefore, the probability of player one winning on the i_{th} round is $(p_1p_2 + (1-p_1)(1-p_2))^{i-1} p_1(1-p_2)$. This implies that the probability of player one winning during sudden death is the following.

$\sum_{i=1}^{\infty} (p_1p_2 + (1-p_1)(1-p_2))^{i-1} p_1(1-p_2)$

$= p_1(1-p_2) \sum_{i=1}^{\infty} (p_1p_2 + (1-p_1)(1-p_2))^{i-1}$

Because both p_1, p_2 are probabilities we have that $0 \leq p_1, p_2 \leq 1$. By looking at the critical points of the expression $p_1p_2 + (1-p_1)(1-p_2)$ we can see that $0 \leq p_1p_2 + (1-p_1)(1-p_2) \leq 1$. However, it is only possible for this expression to be 1 if $p_1 = p_2 = 1$. In this case, the $p_1(1-p_2)$ expression will go to zero and our entire probability will go to zero. Therefore, we can treat our infinite series as converging and it won't cause any issues in our final expression.

We know that $\sum_{i=1}^{\infty} x^{i-1} = -\frac{1}{x-1}$ when $|x| < 1$

Therefore,

$= p_1(1-p_2)\sum_{i=1}^{\infty}(p_1p_2 + (1-p_1)(1-p_2))^{i-1}$

$= \frac{-p_1(1-p_2)}{p_1p_2+(1-p_1)(1-p_2)-1}$

$= \frac{p_1(1-p_2)}{1-p_1p_2-(1-p_1)(1-p_2)}$

Final Probability of Player One Winning

Therefore, the final probability of player one winning is the probability of player one winning in the first five rounds plus the probability that they enter sudden death and win during sudden death.

$P(Player\ One\ Wins) = \sum_{k=1}^{5}\binom{5}{k}p_1^k(1-p_1)^{5-k}\sum_{l=0}^{k-1}\binom{k-1}{l}p_2^l(1-p_2)^{5-l} + \sum_{k=0}^{5}\binom{5}{k}^2(p_1p_2)^k((1-p_1)(1-p_2))^{5-k} * \frac{p_1(1-p_2)}{1-p_1p_2-(1-p_1)(1-p_2)}$

Optimal Banking Strategy

In order to find an optimal banking strategy, we set the problem up as a dynamic programming problem with the following functional equation.

Assume $f(n, r_i, p_j)$ gives the expected return for the optimal banking strategy with n questions remaining, k players, current reward r_i and current player with probability p_j of answering a question correct.

$i \in \{0, 1, 2, ...9\}$
$j \in \{0, 1, ..., k-1\}$

$r_i \in \{0, 20, 50, 100, 200, 300, 450, 600, 800, 1000\}$
$0 \leq p_j \leq 1 \; \forall j$
$\mathbb{1}_{[i+1=10]}$ is an indicator function that equals one if $i + 1 = 10$ and 0 otherwise.

$$f(n, r_i, p_j) = max \begin{cases} p_j f(n-1, r_1, p_{(j+1 \ mod \ k)}) + (1-p_{j)} f(n-1, r_0, p_{(j+1 \ mod \ k)}) + r_i \\ p_j (f(n-1, r_{(i+1 \ mod \ 10)}, p_{(j+1 \ mod \ k)}) + \mathbb{1}_{[i+1=10]} * r_9) + (1-p_{j)} f(n-1, r_0, p_{(j+1 \ mod \ k)}) \end{cases}$$

The functional equation finds the max between the two cases of whether or not the player chooses to bank or chooses to answer the question without banking. The top expression is the case where they bank, and the bottom is when they don't bank. We include an indicator function in the first addend of the bottom expression. This makes it so that if the players reach 1000, the money is automatically banked.