Optimal Strategy for The Weakest Link

Aaron Kruchten, Kaushik Krishna Pradeep, Maximillian Zhang

Introduction

For this project, we attempt to derive and optimal player strategy for *The Weakest Link*. The original version of the show began in 2000 in England and continued to run till 2014; there are over 44 international versions of the show, making it the 2nd most popular TV game show franchise. Although there are slight variations in the various versions, we go with the specifications from the British version as our set-up. 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 question wins. In the event of a tie, the game goes into 'sudden death' - 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 compute the probability a player will win in the final 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.

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.

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

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 $p_1 p_2 + (1-p_1)(1-p_2)$. Therefore, the probability of player one winning on the i_{th} round is $(p_1 p_2 + (1-p_1)(1-p_2))^{i-1} p_1 (1-p_2)$. This implies that the probability of player one winning during sudden death is the following.

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

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

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 $p_1 p_2 + (1-p_1)(1-p_2)$ 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.

Empirical Results for Voting Strategy

Through developing our framework, it became intuitively clear that the optimal voting strategy to maximize a player's expected winnings is dependant on player probabilities. Furthermore, we hypothesized that the optimal voting strategy would also depend on the spread of the player's probabilities. Specifically, consider a game with three players with player probabilities, $p_1 = 0.1, p_2 = 0.8, p_3 = 0.9$. Consider the optimal voting strategy of player three. Although voting off player two would certainly maximize player three's chances of winning the game, the expected winnings in this scenario, according to the formulas and methods outlined above, is actually lower than if player three voted off player one. Thus, in this section, we will outline the methodology and approach we've utilized to examine this problem, showcase the results of our experiments, and provide a road map for future exploration into this subject.

We present a figure that highlights the point of interest. The results are the expected values of dollars earned in the next round given two player probabilities.

Where the expected winnings is computed as follows:

$\mathbb{E}(Expected \ P_1 \ Winnings) = Expected \ Banked(P_1, \ P_2)$ * $\mathbb{P}(P_1$ wins)

Scenarios	P_1 vs P_2	P_1 vs P_3	P_2 vs P_1	P_2 vs P_3	P_3 vs P_1	P_3 vs P_2
S_1	15.83	10.41	30.135	17.717	38.885	33.51
S_2	1.416	0.004	44.962	2.651	88.135	82.072
S_3	5.635	0.004	21.458	0.652	88.135	84.478
S_4	0.047	0.004	75.18	58.52	88.135	151.42

S_1: $p_1 = 0.4, p_2 = 0.5, p_3 = 0.6$,
S_2: $p_1 = 0.1, p_2 = 0.5, p_3 = 0.9$,
S_3: $p_1 = 0.1, p_2 = 0.2, p_3 = 0.9$,

S_4: $p_1 = 0.1, p_2 = 0.8, p_3 = 0.9$.
The results are computed under the assumption that there is zero dollars in the bank.

The figure utilizes the player expected winnings methods listed above to compute the expected earnings in certain scenarios. Due to limitations in time, we've chosen to examine some select cases. As you can see, although the optimal voting strategy is to vote off the player with the highest probability in scenarios one to three; In scenario four, it was optimal for player three to vote off the player with the lowest probability. We wanted to explore this phenomena further to discover the threshold values at which, the voting strategy changes. We first fix one player's probability, we then vary the other two players' probabilities, we compare the expected winnings between the two voting strategies for the highest player, and finally, we record the optimal voting strategy.

The results are presented in the figure below.

p_1	0.1	0.2	0.3	0.4	0.5
Threshold Value Pair	(0.7,0.9)	(0.8,0.8) (0.9,0.7)	(0.8,0.7) (0.9,0.6)	(0.8,0.6)	(0.6,0.6)

The results are listed in ordered pairs $(a, b) = (p_2, p_3)$. Note that the results imply that any combination of player probabilities higher than the threshold pair will result in the optimal voting strategy to be to vote off the player with the lowest probability.

Although the empirical results presented in this section gives us some insight in how the voting strategy dynamically adapts to different player probabilities. The limited amount of data points, due to the high computational cost of running simulations, isn't enough to substantiate any conclusive statement about the optimal voting strategy. Given more time, the empirical approach definitely has the potential to provide a more general understanding of the optimal voting strategy.

Conclusion

Throughout this analysis, we notice that much of the results are intuitive until we backtrack into the earlier rounds. Due to the nature of the dynamic programming framework, earlier rounds result in complex and often intractable solutions.

Some next steps include exploring variations of the parameters for the banking strategy and voting strategies. An additional approach that can be taken is to do the same type of modeling presented in the previous section with parameters that we see during real runs of the game show.