Markov Chain Model for Casino Profit Maximization

Author: Shaoyi Li, Pedro Duarte Moreira, Astrid Zhao, Mofei Wang
APM 115: Mathematical Modeling

Abstract

This article proposes and analyzes a model to determine an optimal slot machine winning rate for a casino to maximize its profits. Due to the stochastic nature of the processes involved, a Markov chain model based on the Gambler's Ruin was used. The optimal winning rates were visualized and computed under different scenarios through both an analytical solution and Monte Carlo simulations, varying the values of several parameters in the process. The results confirm the pattern we would expect in real life.

1. Introduction

In their most basic form, casinos are places in which games of chance are played, and bets are placed on their outcome. By definition, games of chance are games in which there is some randomized process involved. The probabilistic nature of games of chance, the large number of daily players in a casino, and the use of real money makes the use of mathematical models both applicable and useful for analyzing the functioning of casinos. Thus, in this paper we seek to find a simple and reasonable model for a casino to decide the optimal gambler's winning rate to maximize their profit, assuming they can change the winning rate.

In the context of casinos, slot machines are one of their highest earners, comprising about 70% of the income of an average US casino [1]. Also, what distinguishes them from other casino games is that they are completely dependent on chance, meaning that there is no skill or any other factor that could interfere with the outcome, and that casinos then have complete control over the gambler's winning rates. We will then focus on slot machines, and will use a markov chain to model a gambler-based approach to the use of slot machines in casinos.

Markov chains are a type of stochastic model in which the next state of the system depends uniquely on its current state, with the relation between those two states being determined through some probabilistic process. In this study, we will use a variation of the Gambler's Ruin model, which uses a markov chain to model the behavior of a gambler in a game with a given winning rate, and clear "losing" and "winning" states, which when reached by the gambler, signifies that they will stop playing the game. The largest modification is the exclusion of those states, using idle states in their place, meaning that, if the gambler stops playing, it is only momentary. Therefore, in this report we will use a modified Gambler's Ruin markov chain to model a player-based approach to slot machines, and will use it to determine if we can tune the winning rates to maximize profit and if so how, under a variety of assumptions and parameter values.

2. Model

2.1 General Assumptions

In our slot machine gambling model, we make an essential assumption in regards to the functioning of those machines: the player can either lose the money they spent to activate the machine, or get back exactly twice what they put in the machine. Thus, there are only two outcomes: either win or lose x dollars, where x was the amount of money put in the machine. For the sake of simplicity, and because most slot machines only accept one predetermined value, we will consider that predetermined value to be $1.00. We also assume that casinos have total control over the probability of a game being a win or a loss, which is reasonable considering that most slot machines today are digital and thus programmable.

We are considering that players will not play forever, but rather that they will play until they either lose all of their tokens or win a certain amount of money during a session, and then will wait some time before returning for another session. For the sake of simplicity, we assume that all players start playing the slot machines with the same predetermined number of tokens, and that whenever they get more tokens to play (after losing all of their tokens), they get that same amount again. Similarly, if they end up with more tokens then they originally had after the end of a session, we consider that they will convert them to real money before the start of the next session, and will start with that same predetermined amount of tokens again. We also consider that players have an infinite money source.

2.2 General Model

In its most general form, we can represent the model with the following markov chain.

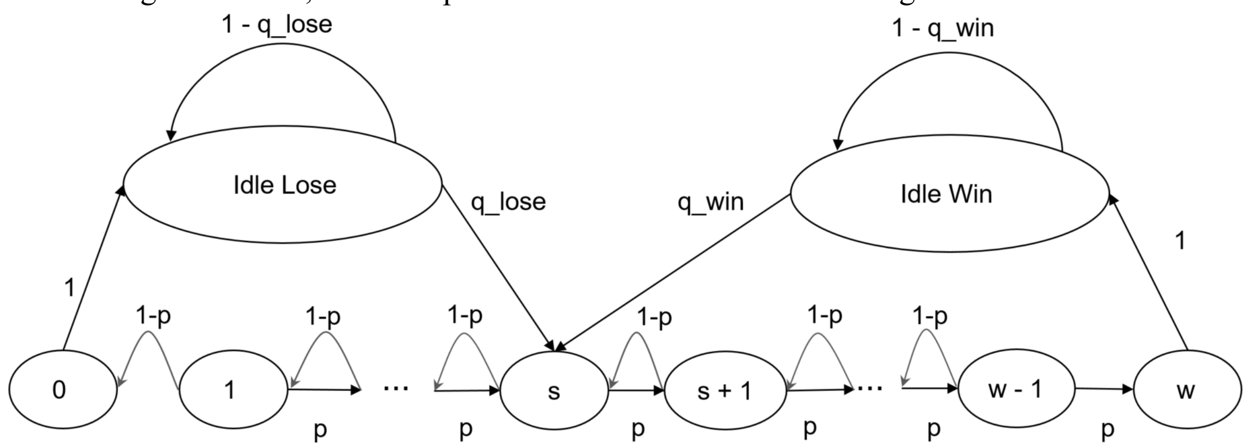

Figure 1. General Markov chain model.

As we can observe from the model, we have a total of w+3 states, where the states numbered from 0 to w correspond to the states that indicate that the customer is playing the slot machines, while the Idle Lose and Idle Win states represent the moments in which the customer is not playing. The way we will refer to w and s throughout the rest of the report is max_money and start_money, respectively, as those were the names they received in the code, but there was not enough space to include the full names in the diagram. At whichever state that is determined by a number, that number represents how many tokens the gambler has in hand at that moment.

We then have that if a gambler reaches 0 tokens, they will go into the Idle Lose state, and that if they win max_money - start_money tokens, they will go into the Idle Win state. As we have discussed previously, after the player returns to the games, their amount of tokens lost and won in a session resets, and they will once again win and lose money until they either win max_money - start_money dollars, or reach 0 tokens. This behavior is possible by having the players start in the start_money state, and return to the start_money state after leaving the idle states. It is interesting to note that the ratio start_money / max_money determines how much risk the gamblers are willing to take, since as that ratio increases, players become less willing to keep playing after they have gained some money. Also, it determines how much money they are willing to lose, in order to gain some money, which is another way of assessing risk.

The rationale behind is that there is a limit in how much players are willing to risk at once. If a player wins a satisfactory amount of money in a session, they will decide that there is far too great a risk to keep playing the game, and lose the money they amassed. Nevertheless, since gambling is addictive, players will inevitably return for another session, and repeat the cycle once again, even if they have to buy more tokens or borrow money, or if they could potentially throw all their winnings away. Thus, there are no absorbing states, meaning that monte carlo simulations are more suitable for determining the behavior of the model (as we will see; however, there are some analytical results, but they involve more complicated mathematics). Moreover, it is noteworthy that the 0 and max_money states are not necessary from a modelling perspective, but they are useful in order to visualize the behavior of the markov chain.

In regards to the numbers p, q_lose and q_win, those indicate the probability of winning the game (winning rate), the probability of going back to the slot machines after losing some money, and the probability of going back to the slot machines after winning some money. To calculate q_lose, we introduce another parameter which is q_lose_ratio, and q_lose is calculated by the following formula: q_lose = q_lose_ratio*p. The rationale behind this is that, as the probability of winning a game decreases, so will the probability that a player will come back for more after losing all their tokens, as they will be more frustrated than a player who is in the same situation, but playing a more winnable version of the slot machine. In this sense, q_lose_ratio indicates how more frustrated players become by playing less winnable slot machines. This of course assumes that individual players will either figure out the winning rates fairly quickly or that they will collectively find it by communicating with each other. Although those assumptions may not be completely reasonable, introducing player frustration in a more complex manner would make the model too complicated.

To obtain the profit per player, we perform a monte carlo simulation of the model, with some number of gamblers n_gamblers and some number of steps n_steps. Profit is calculated by adding \$1.00 to the profit whenever a player loses a game, and subtracting \$1.00 whenever they win a game. Thus, we are considering that one token is worth \$1.00, for the sake of simplicity. To get the profit per player, we take the mean of all the profits generated by individual players. For this model, we assume that all players will spend the same total amount of time in the casino, and that varying the other parameters does not necessarily have an effect on the number of gamblers. Once again, those assumptions are not completely accurate, but they are not completely misguided either, and taking more complex behaviors into account would make the model overly complicated.

2.3 Default Parameter Values

In regards to the baseline or default model, the major assumptions we make are that max_money is 6, and that start_money is 2 for this scenario (that value will be varied in the analysis section, however). Since there are not many parameters in our model, we can change how the modelled system responds to variations of most parameters inside a reasonable spectrum, and there is no baseline value for those parameters. Our analysis section includes studies in how the model responds to variations in start_money, p, q_lose_ratio, and q_win. From all of those values, p is the only one that the casino can directly affect and thus every simulation involves some variation of it to find the maximum profit. The other parameters, s, q_lose_ratio and q_win are more environmental, and although they could possibly be indirectly affected by the casino, they would likely be found through empirical methods. Nevertheless, some variation of those parameters is also taken into account to see how the system behaves under different conditions, and to make the model more general, as those values can vary from casino to casino, and are not easily determinable. Here is the canonical form of the markov chain transition matrix for our baseline scenario.

$$A = \begin{array}{cc} & \begin{array}{ccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & I_{lose} & I_{win} \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ I_{lose} \\ I_{win} \end{array} & \begin{pmatrix} 0 & 1-p & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1-p & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & p & 0 & 1-p & 0 & 0 & 0 & p * q_{lose\ ratio} & q_{win} \\ 0 & 0 & p & 0 & 1-p & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & p & 0 & 1-p & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & p & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & p & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p * q_{lose\ ratio} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1-q_{win} \end{pmatrix} \end{array}$$

The only parameters that are consistent throughout our analysis are n_gamblers and n_steps. As for n_gamblers and n_steps, variations in n_gamblers have diminishing returns, as with a sufficiently large number, we get fairly close to the expected value with simulations, and after that point simulations would become larger with no useful reward in accuracy. Its value is then fixed at 5000. The same is true for n_steps, as after a certain point the average profit per time would become very close to the mean, and having those two parameters fixed at some value makes it easier to have consistent results. The value for n_steps used throughout the analysis section is 240.

It is important to note that we did not factor in the costs related to machine operations, repairs, etc. in the calculation of profit, as those are generally independent of our model.

Below is the transition matrix in Python with p=0.4 and default parameter values, with state labels:

	0	1	2	3	4	5	6	idle_lose	idle_win
0	0.0	0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.6	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.4	0.0	0.6	0.0	0.0	0.0	0.1	0.5
3	0.0	0.0	0.4	0.0	0.6	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.4	0.0	0.6	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.0
idle_lose	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.9	0.0
idle_win	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.5

3. Analysis

To compare the maximum profit under different p values, we are going to further set up parameters with default values. Under our basic model, we have q_lose_ratio = 0.25, q_win = 0.5, start_money = 2 and max_money = 6. In the sensitivity analysis below, we are going to test sensitivity on those parameters.

3.1 Sensitivity analysis

For the figures below, we are trying to find the maximum profit by varying p value, so the x-variable would be p and y-variable would be profit. The curves would first intersect at (0, 2) since the gamblers would have a 100% chance to lose their first and second round, and they will never want to come to the casino again. The next intersection would be at (0.5, 0) because the expected profit for the casino becomes 0 as the winning probability gets to 0.5. (calculation shown in Appendix, Scenario 1) Thus, in the standing point of the casino, we definitely pursue making money and make p always smaller than 0.5.

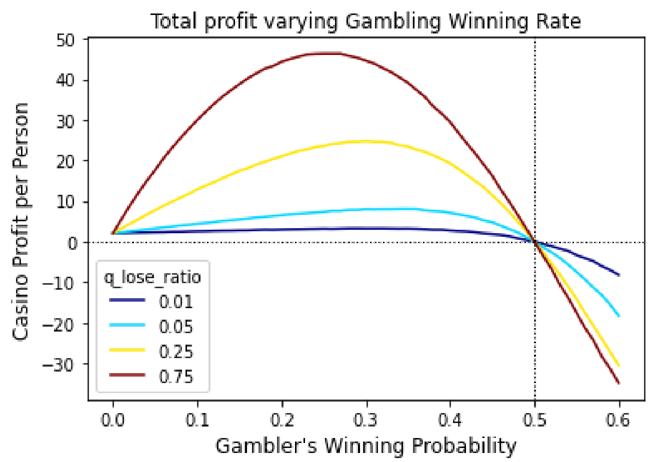

Figure 2

In Figure 2, the four curves indicate different q_lose_ratios. Clearly, when the q_lose_ratio is higher, the maximum profit gets higher. Since if the q_lose_ratio is higher, people in the idle lose state are more willing to get back to the casino. Thus, another way to maximize the profit is trying to make gamblers play more rounds instead of staying in the idle state of either win or lose. At the same time, since people are more willing to get back to the casino after losing if the q_lose_ratio increases, we want to make p value relatively small to make sure they lose faster. That means an increase of q_lose_ratio could shift the maximum profit to the left.

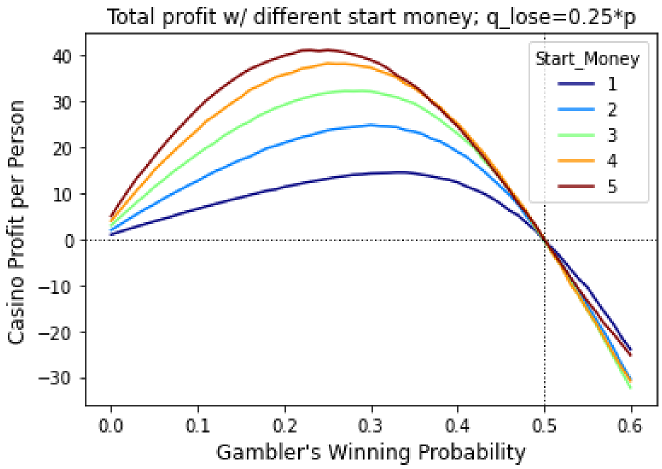

Figure 3

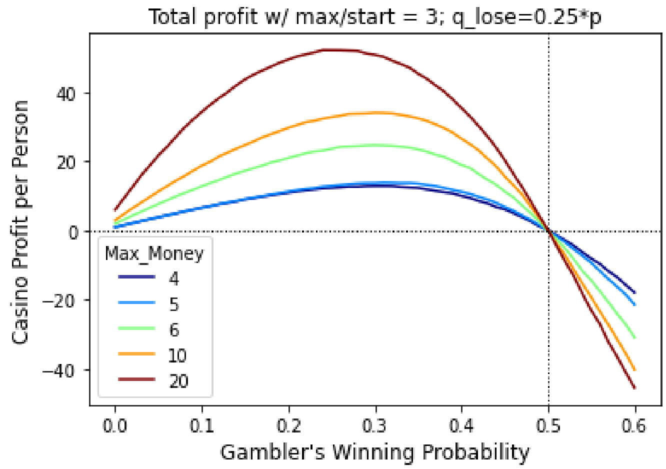

Figure 4

Figure 3 shows the profit with different start_money and Figure 4 shows the profit with different sets of max/start money. It is obvious that the highest start_money gives us the highest maximum profit because people bring a higher amount of money to the casino (and they are always going to lose!). We can also see start_money has a larger impact than max_money. Generally, as the start_money increases, the maximum profit point shifts to the upper left with smaller p value. We want to make gamblers have a lower chance to win their first round and we do not want gamblers to stay in idle lose state for too long.

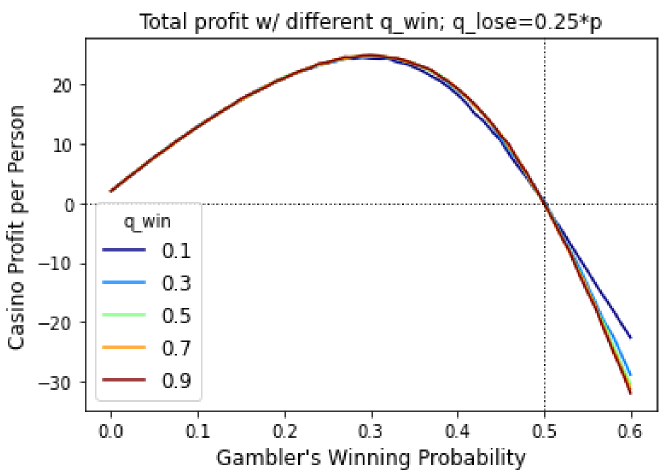

Figure 5

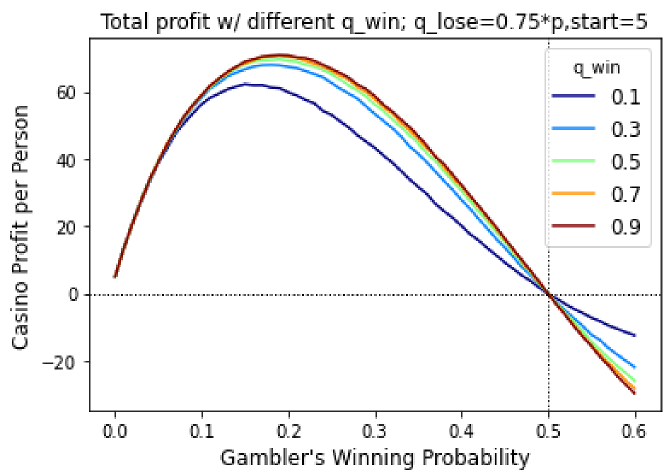

Figure 6

The figures above show the profit with different q_win. In Figure 5 we can see all curves overlap together before p value reaches 0.3. The reason is all gamblers with start_money = 2 have less than 3% chance to get to the max_money = 6; therefore, q_win has a really small impact in this case (mathematical derivation shown in Appendix, Scenario2). To make a clearer contrast, we set q_lose_ratio to 0.75 and start_money to 5 in Figure 6. In this case, there is a much higher chance that people will win the first round and go to idle state of win (higher q_lose_ratio will make people have a lower staying time in idle state of lose, which will increase the impact of q_win). Higher q_win indicates higher maximum profit, and it also shifts the maximum profit to the right, because people will be more likely to come back to the casino.

3.1.1 Figure summary

Looking at Figure 2 to Figure 6, we can easily conclude they all have similar patterns for profit varying p and more importantly they have maximum points on all curves with p somewhere between 0 and 0.5. When p equals 0.5 (perfectly fair game) all curves have the same value 0 -- which is intuitive. When p is greater than 0.5 the average profit becomes negative no matter what. When p equals to 0 the curves basically all have the value of start_money which is the scenario that the gamblers lose all the money they start with and never come back from Idle Lose.

3.2 Extension: analytical solution

While monte carlo simulation is very helpful in generating the distributions of the analytics of interest (the casino's profit in our case), the computational power is still a big limitation if we want to conduct more extensive analysis given the resources we have. Considering here we focus on the mean of the simulated distribution, it's worth trying to see if we can derive the analytical solution for the mean of the profit distribution.

Derived close-form equation for casino's profit with default parameter value set q_lose_ratio = a, q_win = 0.5, start_money = 2, max_money = 6, the expected total profit per gambler is :

$$E_{tp} = (2 - 6w(p))\left(\frac{\left(\frac{6w(p)}{1-2p} - \frac{2}{1-2p} + 240\right)}{w(p)\left(-\frac{1}{ap} - \frac{6}{1-2p} + 2\right) + \frac{1}{ap} + \frac{2}{1-2p} + 1} + 1\right)$$

with

$$w(p) = \frac{(\frac{1-p}{p})^2 - 1}{(\frac{1-p}{p})^6 - 1}$$

Complete solutions and derivations are shown in the Appendix.

3.2.1 Comparison and verification: analytical vs simulated

With the default parameter value set, we plot the profit varying p curves using both the analytical solution and the simulated solution. The curves look very consistent, as expected.

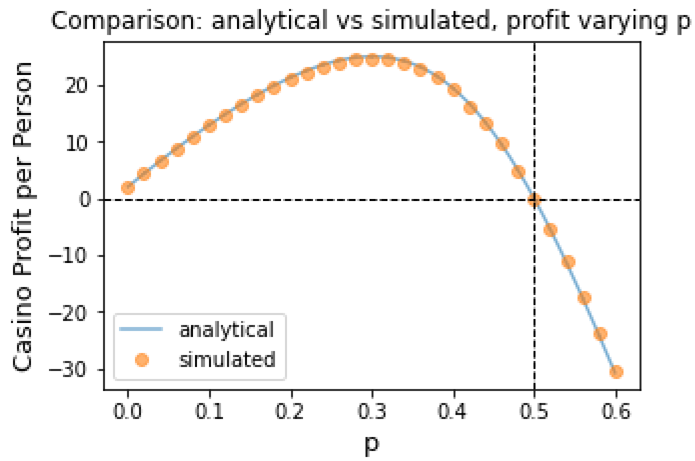

Figure 7

3.2.2 Extensive sensitivity analysis

The Figure 2 in 3.1 gives us an idea on the profit patterns varying p with 4 different q lose ratio values. The curves on that figure have different maximum values at different values of p. How does the maximum point location change when we use a broader selection of q lose ratios? Could we find some patterns?

For simplicity, we define option p as the value of p when mean profit reaches maximum with fixed q lose ratio. In the left figure below we can see max profit grows monotonically, with a slowing rate, when q lose ratio increases -- which is pretty intuitive.The right figure below interestingly shows a maximum when q lose ratio is 0.05. This is something difficult to be captured if we only use monte carlo simulation with limited computational resources. It means the optimal p increases together with q lose ratio when q lose ratio is below 0.05 and then starts decreasing. In the language of the casino, when q lose values are very small, a relatively larger q lose value scale leads to a better winning probability for its customers for maximum profit; but when q lose values are big enough, a relatively larger q lose value scale leads to a lower optimal winning probability.

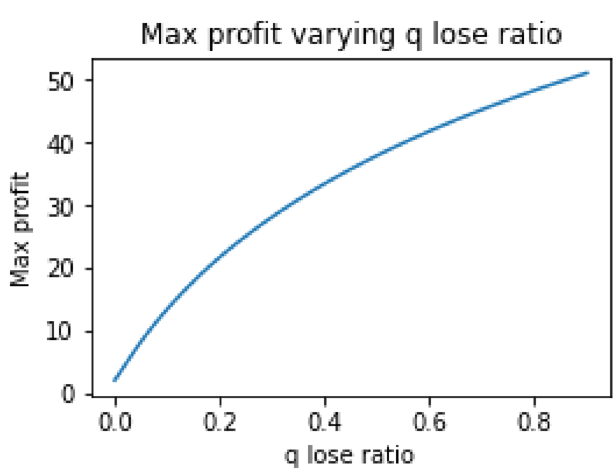

Figure 8

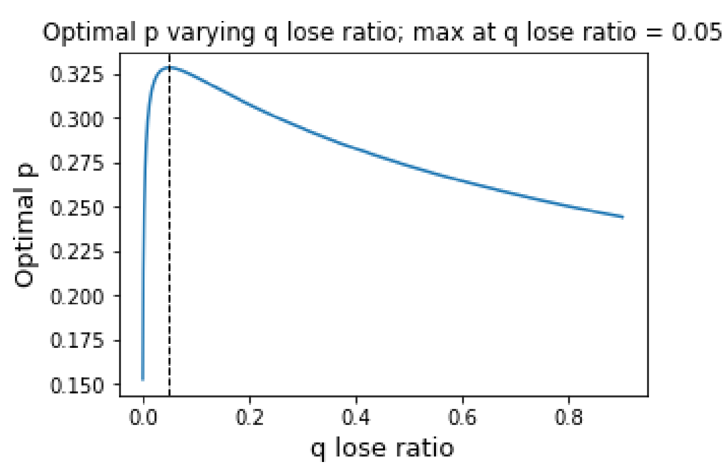

Figure 9

4. Discussion

4.1 Findings and summary:

We used Monte Carlo simulations because it is intuitive, relatively easy to implement without getting into the mathematical complexity. The model simulates what would be happening in real life and it is easy for us to change parameters around to test the sensitivity. From Figure 7, we can see the results are consistent with analytical solutions with enough realizations (in our case, enough gamblers). The largest con here is it takes a lot of computing power and that's why the deterministic model sometimes helps. In addition we also try to start with a simple model to limit the number of moving pieces and make the model stay intuitive when parameters are being changed.

Our goal is to maximize the casino profit, and in order to get the highest profit, we need to find an optimal p value. If p value is too low, people would stay in idle lose state for too long, and if p value is too high, our expected profit per game is going to be lower. Also, the p value is always less than 50%, then the casino will always make a profit in the long run. From our analysis above, the optimal p ratio falls between 0.15 to 0.35.

Gamblers who won some money are usually more willing to come back to the casino, so our q_lose should always be less than q_win. The expected rounds to stay in idle lose state is 1/q_lose and the expected rounds to stay in idle win state is 1/q_win. When the gamblers are greedy (the max_money is far away from start_money), q_win doesn't matter that much because most of the gamblers will go to the idle lose state. When gamblers are self-restrained, larger q_win gets us more profit and it shifts the maximum profit point to the right. Generally, higher q_lose_ratio or high q_win will get us more profit.

On the other hand, start_money and max_money also play a big role. If gamblers bring in more money, the amount of time they spend in the casino will increase and then the profit increases. In real life, the max_money depends on the gamblers' desire for money.

To conclude, to get the maximum profit, we want the gamblers to spend as much time as they can in the casino. We can increase q_lose_ratio, q_win, start_money and max_money by advertisement, better service, and game rule setting. Then we should collect the data and find the optimal p value. It is fine if the gamblers win some games, but the profit will definitely decrease if less people want to play!

4.2 Limitations and further extensions:

- Gamblers in theory can win unlimited amount of money and they can leave the table to be idle with any money -- does have to reach 0 or max money
- q win can also be a function of p too
- q lose doesn't have to be proportional to p. In practice, people tend not to notice the change of p when p is close to 0.5 but if p becomes closer to 0 gamblers will turn frustrated at the game very soon as p decreases.
- q lose can also be path dependent as in practice it's driven by the actual experience of the past games -- this is actually one scenario which is difficult to model with Markov chains.x
- When gamblers win and become idle, they are very likely to consume inside the casino so the model can take account of profit made in this idle state.

- We assume we only care about the means of profit made by the casino. However, in reality the left tail of the profit distribution is important too. Given casino is usually a big business so it worries less about the tail case where one day it loses a lot of money -- however it's still something to be considered in making corporate decisions for a mature business.
- Again, one of the downsides of monte carlo simulations is the computational cost. Given the resources we have we decided only to simulate each gambler for 240 rounds (time steps). We should be able to visualize other more granular parameter sensitivity and get higher resolution results from the simulated distributions with more resources.

5. Conclusion

Like most other businesses, one important nature of casinos to consider is profit maximization. It's interesting to discuss how can a casino find the optimal winning probability for the gamblers in order to maximize the profit sustainably, assuming they are capable of freely setting the winning probability. A Markov chain model with monte carlo simulation is used to analyze the problem. To simulate the possible strategies, we create different scenarios by varying primary parameters such as start money, max money, or the probability of gambler will restart the game from the idle state, which aim to find when gamblers could have continuous willingness to play. The model suggests that to maximize profit, we need to increase gamblers' willingness to play the games, or in other word, we need to increase their time spent in the casino. It means we want to increase gamblers' start_money, q_win or q_lose, and this can be done by good service and efficient advertisements that make gamblers believe they can win the game. The model further suggests that the optimal winning probability can be found and is usually about 0.15 to 0.35 depending on the other parameters, which is not far from our expectation in reality. At last, analytical solutions and simulated solutions show the same result -- if the probability of winning the game is less than 50%, casinos always make a profit in the long run.

Appendix(analytical solution):

We first define some variables:
q_lose_ratio = a,
q_win = b
max_money = m
start_money = s
number of rounds = r
winning probability = p
We also call the gambler go from idle states back to idle states a loop
We have the expected winning chances for gamblers (win the max_money from start_money):

$$w(p) = \frac{\left(\frac{1-p}{p}\right)^{s} - 1}{\left(\frac{1-p}{p}\right)^{m} - 1}, \quad if \quad p \neq \frac{1}{2}$$

$$w(p) = \frac{s}{m}, \quad if \quad p = \frac{1}{2}$$

The expected losing chance for gamblers would be:

$$1 - w(p)$$

The expected profit for each loop would be:

$$E_p = w(p) * (s - m) + (1 - w(p)) * s = s - w(p) * m$$

Expected rounds from start_money to max_money/state 0 is:

$$E_r = \frac{s}{1-2p} - \frac{m}{1-2p} * w(p)$$ [2]

Expected rounds from max_money/state 0 to start money is:

$$(1+\frac{1}{b}) * w(p) + (1+\frac{1}{ap}) * (1 - w(p)) = 1 + \frac{1}{ap} + (\frac{1}{b} - \frac{1}{ap})w(p)$$

Then the expected total rounds per loop is :

$$E_{tr} = \frac{s}{1-2p} - \frac{m}{1-2p} * w(p) + 1 + \frac{1}{ap} + (\frac{1}{b} - \frac{1}{ap})w(p)$$

We want to ensure that the gambler plays at least one loop, so the expected number of loops would be:

$$\frac{r - E_r}{E_{tr}} + 1$$

The total expected profit for casino would be:

$$E_{tp} = E_p * (\frac{r - E_r}{E_{tr}} + 1)$$

Scenario 1:
When p = 0.5:

$$w(p) = \frac{s}{m}$$

and then

$$E_{tp} = E_p * (\frac{r - E_r}{E_{tr}} + 1) = (s - \frac{s}{m} * m) * (\frac{r - E_r}{E_{tr}} + 1) = 0$$

Scenario 2:
When p = 0.3 and s = 2, m = 6:

$$w(p) = \frac{(\frac{1-0.3}{0.3})^2 - 1}{(\frac{1-0.3}{0.3})^6 - 1} = 2.77\%$$

Scenario 3:
To check the relationship between p and a at maximum profit with our basic setting (b = 0.5, s = 2, m = 6):

$$w(p) = \frac{(\frac{1-p}{p})^2 - 1}{(\frac{1-p}{p})^6 - 1}$$

and then

$$E_{tp} = (2 - 6w(p))\left(\frac{\left(\frac{6w(p)}{1-2p} - \frac{2}{1-2p} + 240\right)}{w(p)\left(-\frac{1}{ap} - \frac{6}{1-2p} + 2\right) + \frac{1}{ap} + \frac{2}{1-2p} + 1} + 1\right)$$

Contribution:

Pedro Duarte Moreira (model idealization, assumptions and relevant processes; interpretation and expectations of model behavior; abstract, sections 1 and 2 of the report; slides);(about 24% of work)

Mofei Wang (model idealization; organized code and generated graphs; section 3, 4, 5 of the report;slides; analytical solution) (about 33% of work)

Shaoyi Li (coding; coordination; model idealization, assumptions and relevant processes; interpretation and expectations of model behavior; sections 3, 4, 5 and overall editing of the report; slides)(about 33% of work)

Astrid Zhao (write the report with section 1, 5, and part of 3, help to polish section 3 of the report, slides)(about 10% of work)

References:

1. Cooper, Marc (December 2005). "How slot machines give gamblers the business". The Atlantic Monthly Group. Retrieved 2008-04-21

2. Leighton, T., & Rubinfeld, R. (2006, December 12). *Random Walks/The time to completion*. https://web.mit.edu/neboat/Public/6.042/randomwalks.pdf.