Monty Hall - Monte Carlo

Ryan McArthur
California State University, Fresno
College of Science and Mathematics
Fresno, United States
merccat@mail.fresnostate.edu

Sarah Hang
California State University, Fresno
College of Science and Mathematics
Fresno, United States
satoji@mail.fresnostate.edu

***Abstract*—This paper explores the Monty Hall and Monty Fall problem through three approaches: the host reveals n-2 doors and the contestant switches on the last door, each time the host reveals a door, the contestant can switch to any door, and the contestant keeps their door. Both variants and their approaches are further expanded to 6, 9, 20, and 100 doors. By simulating the Monty Hall and Monty Fall problem, this study aims to compare the win rate of switching doors and keeping doors.**

***Index Terms*—Monty Hall, Monty Fall, Monte Carlo Simulation, Probability**

I. Motivation

We decided to tackle the Monty Hall problem because it was difficult trying to wrap our heads around there being a better chance of winning a car by switching doors instead of keeping an original selected door. The motivation of Monty Hall is to find and compare the probability of winning by switching doors and keeping doors.

II. Problem Statement

The Monty Hall problem originated from the television game show Let's Make a Deal. A contestant has the option to select one out of three doors shown where two doors contain a goat and one door contains a car. Once the contestant has selected a door, the host always reveals a door with a goat behind it. From here, the contestant has the option to either stick with their door or switch to the other door. Expanding this problem to 6, 9, 20, 100 doors instead of 3, the contestant would have the option to choose one door out of n doors. A variant of the Monty Hall problem, known as Monty Fall, is where the contestant has chosen a door and the host slips on a banana peel and accidentally pushes open another door. If the host accidentally reveals the car, then the trial is invalid and we move on to the next trial.

III. Related work and background

We first encountered the Monty Hall problem in class and learned that switching doors lead to more chances of winning than sticking to the original door. We further explored the Monty Hall problem through various online sources involving the original three doors, n doors, and Monty Fall variant that way we can get an idea of how to approach the problem.

IV. Approach

We created a function to create n number of doors, a function to perform the Monty Hall simulation, and pass in parameters to the Monty Hall function based on the scenario. The parameters passed in are the number of simulations, number of doors, a boolean to switch doors, a boolean to switch doors each time the host reveals a door, and a boolean variable to perform the Monty Fall variant. Within the Monty Hall function, we calculate the win and loss probabilities based on the variants. At first we implemented Monty Hall as if we were a contestant ourselves where we had the option to stick or switch doors. We then modified our code to perform the Monte Carlo simulation and randomly select a door instead of manually selecting a door.

The three options that we implemented of Monty Hall for n doors include allowing the contestant to switch to the last unopened door at the end, switching doors after each time the host reveals a door with a goat, and keeping the door. Monty Fall also utilizes all three options as well and has different results.

V. Experiment set-up

In the Monty Hall function, we initialize a win counter and loss counter to zero. We then iterate through the function based on the number of simulations passed in, create doors, select a random door. While the number of doors is greater than two, all the doors the host can choose is computed where it either contains all the doors that contain a goat and is not the selected door or in the Monty Fall variant, the host can choose all the doors besides the selected door. In other words, the host can accidentally select the door with a car. The host then reveals one door and we remove the revealed door and change the position of the selected door in the array if the removed door was before the selected door.

In the Monty Fall case, if the revealed door is a car, the trial is removed and continues on to the next trial. In the version where we can randomly switch doors after every reveal, the selected door's position in the array changes. In the other version, we always switch to the other door if the switch parameter passed in is set to True. If the selected door is a car, then we increment the win counter, otherwise we increment the loss counter. The win and loss probabilities are then calculated by the number of wins divided by the number of simulations and the number of losses divided by the number of simulations respectively.

VI. Results and Discussion

We discovered that for the original Monty Hall problem containing only three doors, the probability of winning by switching doors at the end is roughly 0.677 whereas the probability of losing is roughly 0.323 when running 1,000 simulations. In other words there is a 2/3 chance of winning and a 1/3 chance of losing. The probability of winning by switching doors randomly after each reveal gives us a 0.668 compared to losing gives us 0.332. For three doors, switching doors randomly after each reveal is the same problem since we are either forced to switch or stay with the door after the host reveals the door. The case of switching doors randomly after each reveal only affects 4 or more doors. The probability of always keeping the door results in 0.304 win compared to 0.696 loss out of 1000 simulations.

The Monty Fall variant for three doors gives the probability of winning to be 0.488 compared 0.512 by losing. This means that when the host accidentally reveals a goat, there is a 1/2 chance of getting either a goat or a car. For switching a door randomly after each reveal, there is a 1/3 chance of winning because you have one option out of the three to select a door and the host accidentally revealing a door does not make a change in our decision. Similarly, for keeping the door in Monty Fall, our probability of winning is still 1/3 due to only having one option out of three and the host's actions are accidental.

As for 6, 9, 20, and 100 doors, the actions of the host and results are quite similar. For selecting a door, having the host reveal all doors besides the car door, and then switching doors at the end, in the case with 6 doors, the probability of winning is 0.842, 9 doors is 0.901, 20 doors is 0.955, and 100 doors is 0.995. In other words, the more doors the host reveals, the higher the probability of winning and lower the probability of losing are. For switching to a door after every time the host reveals the door, we found the probability of winning and losing was consistent for all n doors where the probability of winning is roughly 0.666 and losing is around 0.333. The probabilities are consistent throughout because the host is always revealing doors and given the process of elimination by switching doors each time, our chance of getting the car would be the same as the three doors case. The probability of always keeping the door for n doors decreases our chances of winning the more doors we have. The probability of keeping the door and winning for 6 doors is 0.173, for 9 doors is 0.115, 20 doors is 0.047, and 100 doors is 0.013. The probabilities appear to be accurate of the situation since the host is eliminating all the doors up until there is a car left.

For Monty Fall with 6, 9, 20, and 100 doors, the probabilities also increase or decrease depending on the options to switch or stick with the original door. The probability of switching doors at the end is 0.358 for 6 doors, 0.292 for 9 doors, 0.17 for 20 doors, and 0.053 for 100 doors. Thus, the more doors there are, the lower the chance is of winning because the host's actions are accidental and the host essentially doesn't know if the door contains a goat or car. As for switching the door after each reveal, the probability of winning is 0.168 for 6 doors, 0.108 for 9 doors, 0.049 for 20 doors, and 0.009 for 100 doors. The probability of keeping the door and winning for 6 doors is 0.181, for 9 doors is 0.097, for 20 doors is 0.048, and for 100 doors is 0.006. We can conclude from the Monty Fall variant that the option to switch or stick doors is random and the more doors there are, the more difficult it is to win for all three options that we implemented.

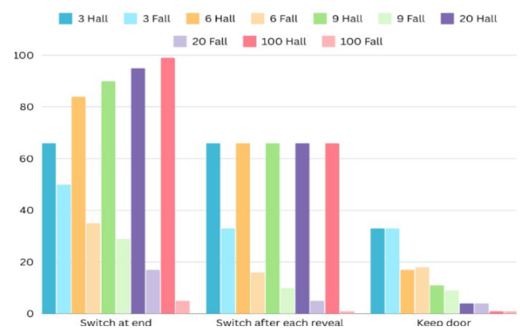

Fig. 1: Results of the 3 options for Monty Hall and Monty Fall. Lighter colors represent the Monty Fall variant.

VII. Contributions and Conclusions

After exploring the Monty Hall problem, variants, and approaches, we can conclude that switching doors often leads to a higher rate of winning compared to sticking with the initial door for all cases of 3, 6, 9, 20, and 100 doors except for the Monty Fall variant. By implementing the Monty Hall problem as a Monte Carlo simulation, we obtain a better understanding of decision making and the optimal strategies for Monty Hall and its various approaches.