Results Report - The Hogwarts Hobo game

Team 29
Adam Sorrenti, #500903848
Kritarth Shah, #500907217
Kateryna Shkinova, #500911953
Keith Jose, #500816528
Raunak Bajaj #500912844

Table of Contents:

Overview:

This report shows the results of our experiment which compared the gameplay performance of a simple algorithm with our optimized algorithm. The report outlines parameter values, duration of the test runs and compares the benchmark results with that of the optimized algorithm. Results indicate that the optimized algorithm performs 20% - 75% better than the simple algorithm under normal (non-extreme) circumstances.

Algorithm Description:

Simple Algorithm:

The simple algorithm moves the player to the track immediately next to the player's current track upon getting hit. If the player is at the last track in the tunnel, the player is moved to the first track in the tunnel.

Optimized Algorithm:

While devising an application to manage the exercise and maximize player lifetime we created an optimized algorithm using a machine learning approach. More specifically, we implemented a tail-recursive algorithm that accumulated a dataset for each track containing the safe time periods for a given game instance. Using this information the algorithm was able to determine the best next track. If not enough information was present to make an informed decision the algorithm chose the next track at random.

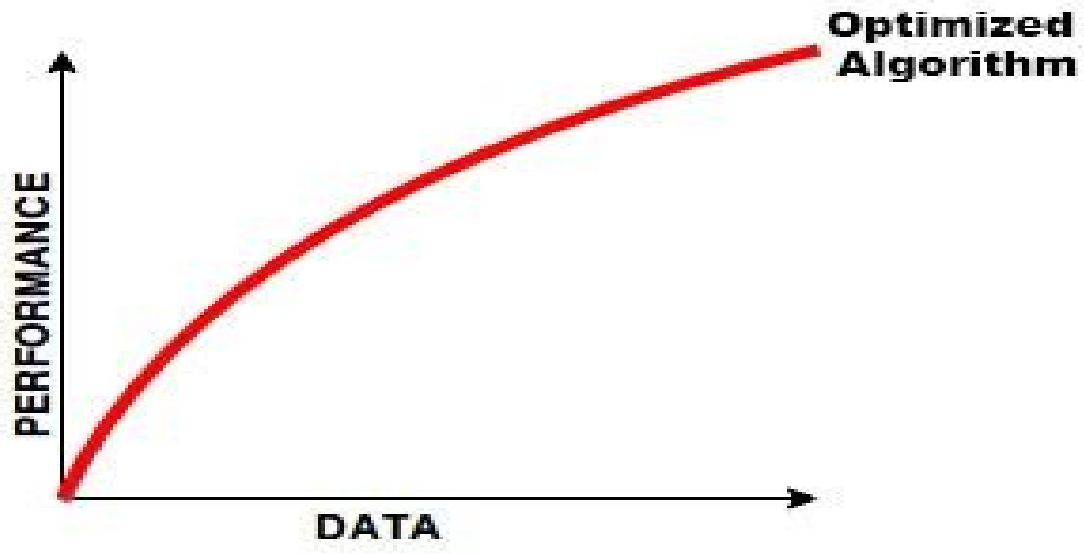

Figure 1: We noticed the following trend in our algorithm; as the amount of data increased so did the performance in-game.

Optimization Algorithm Results:

The following graphs show the results of the multiple tests conducted on both the simple algorithm and the optimized algorithm with varying parameters. The success of the test was determined by whether the optimized algorithm resulted in fewer collisions than the simple algorithm. Each test carried over collision information for the optimized algorithm to use in determining the best track to move to each time. The results are split into normal and extreme cases. It is important to note that sets of tests may yield different results due to the fact that the probability distribution that models the trains is different for each game and thus will also be different between sets of tests.

Normal Cases:

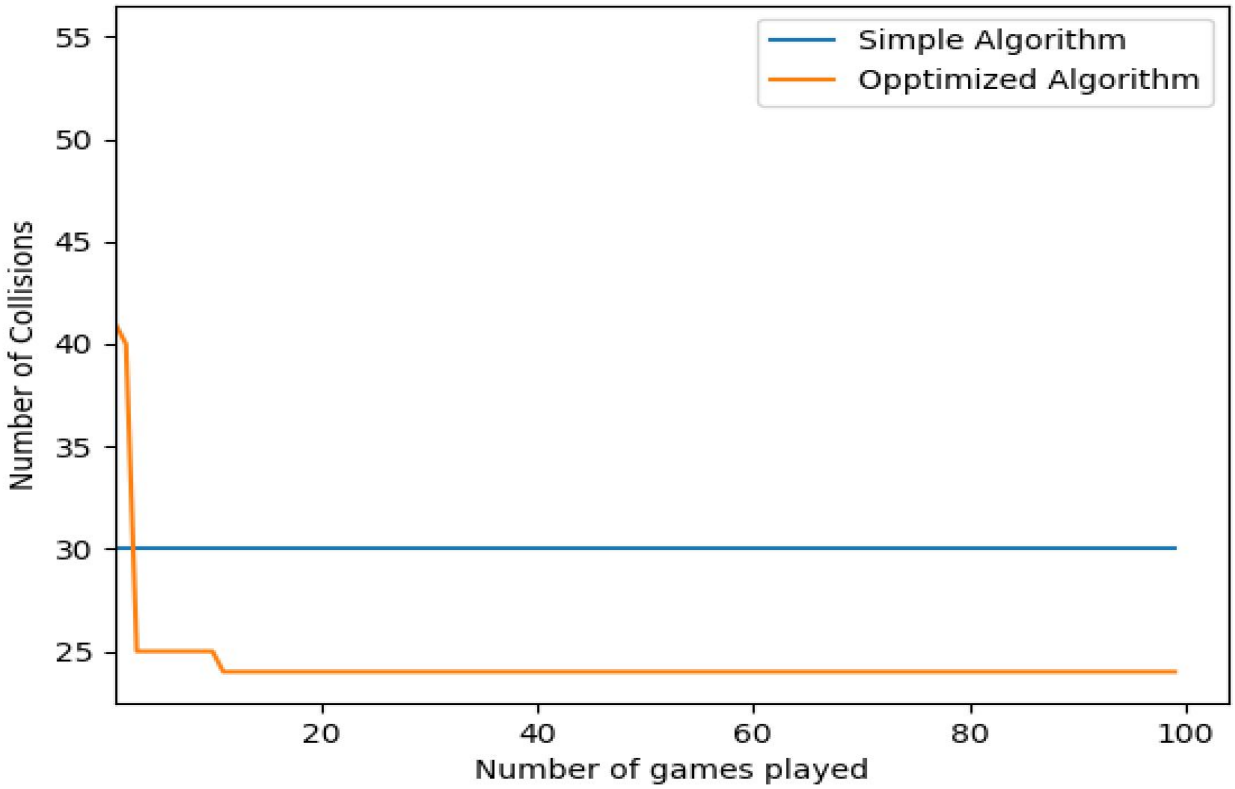

Figure 2:

The parameters used for the testing in Figure 2 are:

M (Number of Tracks) = 10
S (Player Time on Each Track) = 10

Out of the 100 tests conducted, 3 tests failed and 97 tests passed.

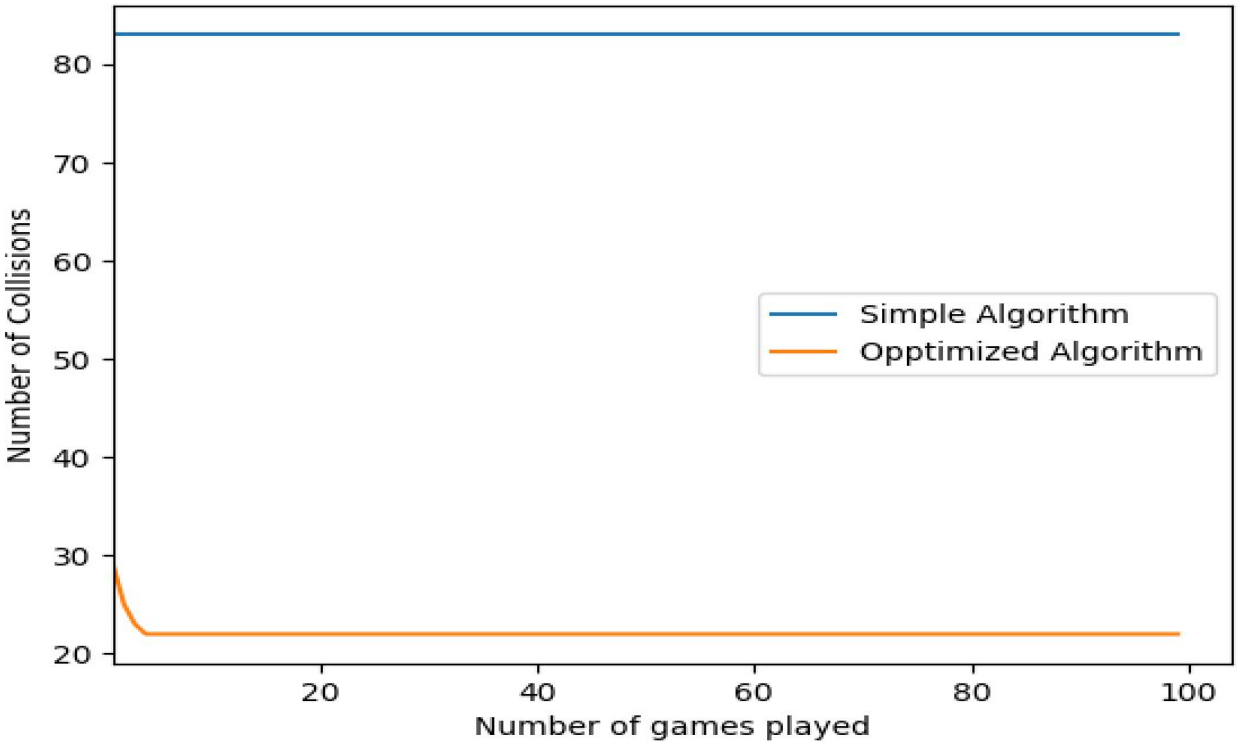

Figure 3:

The parameters used for the testing in Figure 3 are:

M (Number of Tracks) = 10
S (Player Time on Each Track) = 10

Out of the 100 tests conducted, 0 tests failed and 100 tests passed.

Extreme Cases:

Notice the three extreme cases. Each of the parameters include exactly two tracks and an arbitrary amount of time on each track. This results in one out of the three results shown below. Due to the random probability distribution and starting track for the optimized algorithm you cannot be consistently successful when there are only two tracks in a tunnel.

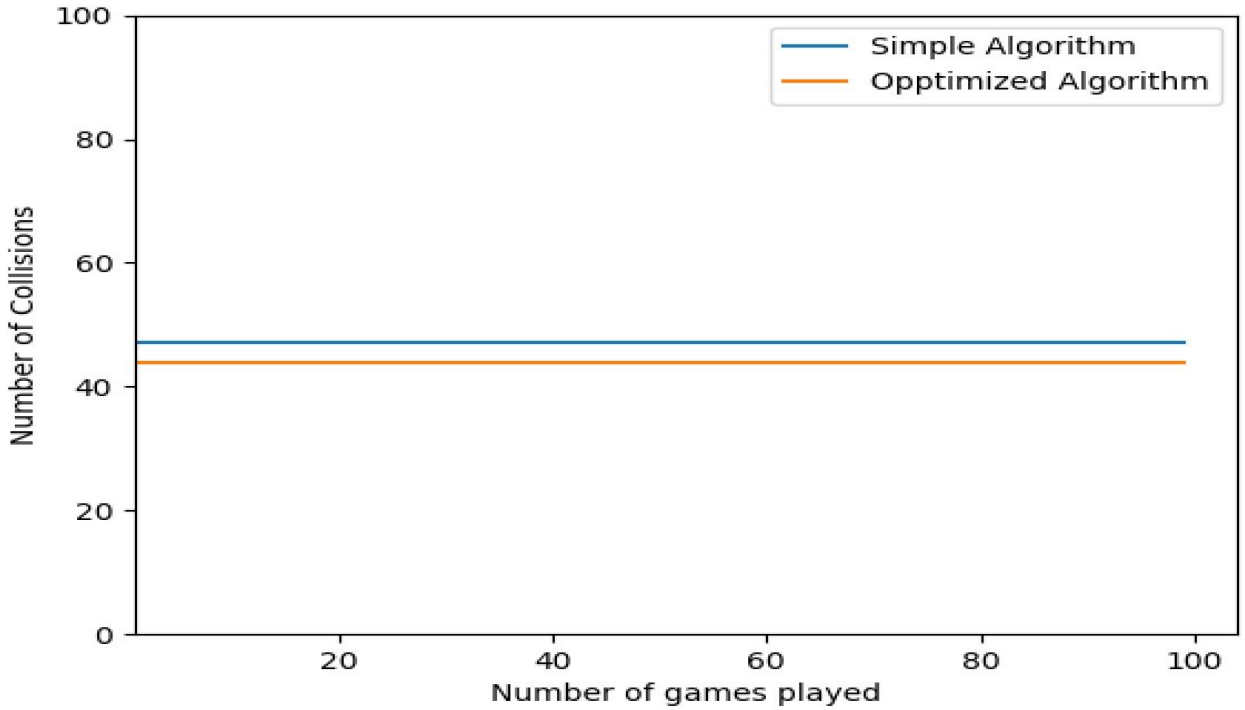

Figure 4:

The parameters used for the testing in Figure 4 are:

M (Number of Tracks) = 2
S (Player Time on Each Track) = 10

Out of the 100 tests conducted, 100 tests failed and 0 tests passed.

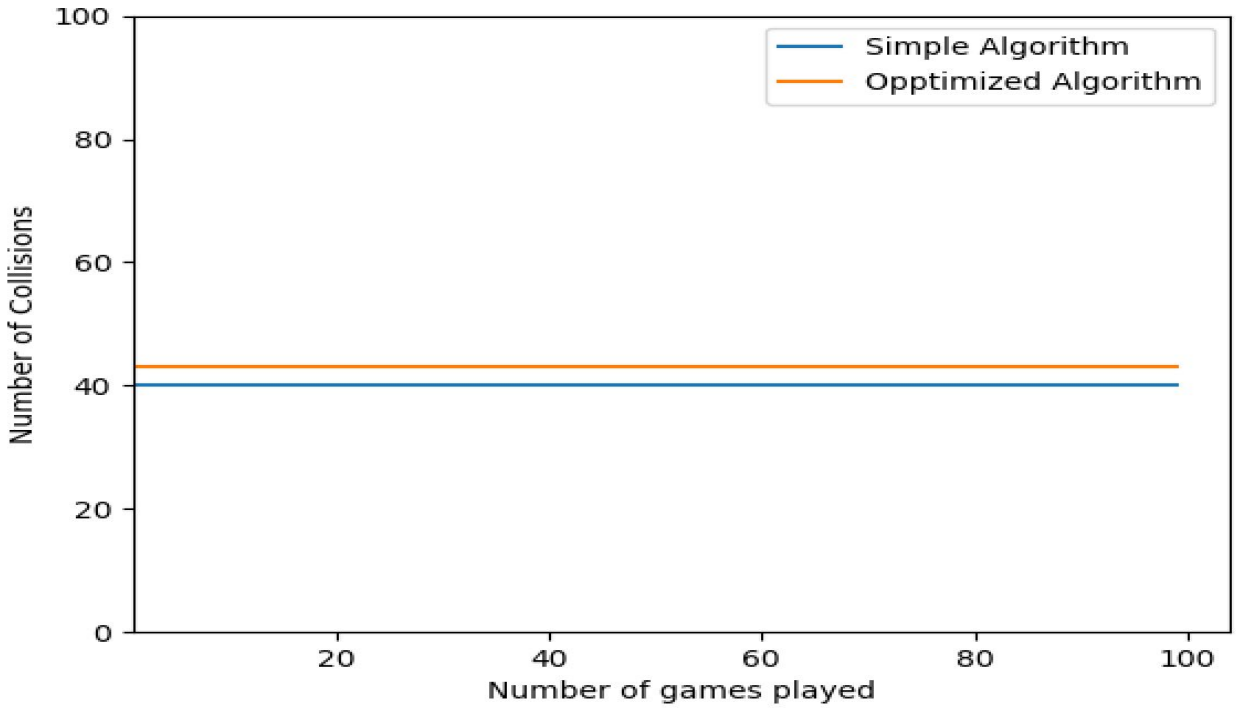

Figure 5:

The parameters used for the testing in Figure 5 are:

M (Number of Tracks) = 2
S (Player Time on Each Track) = 10

Out of the 100 tests conducted, 0tests failed and 100 tests passed.

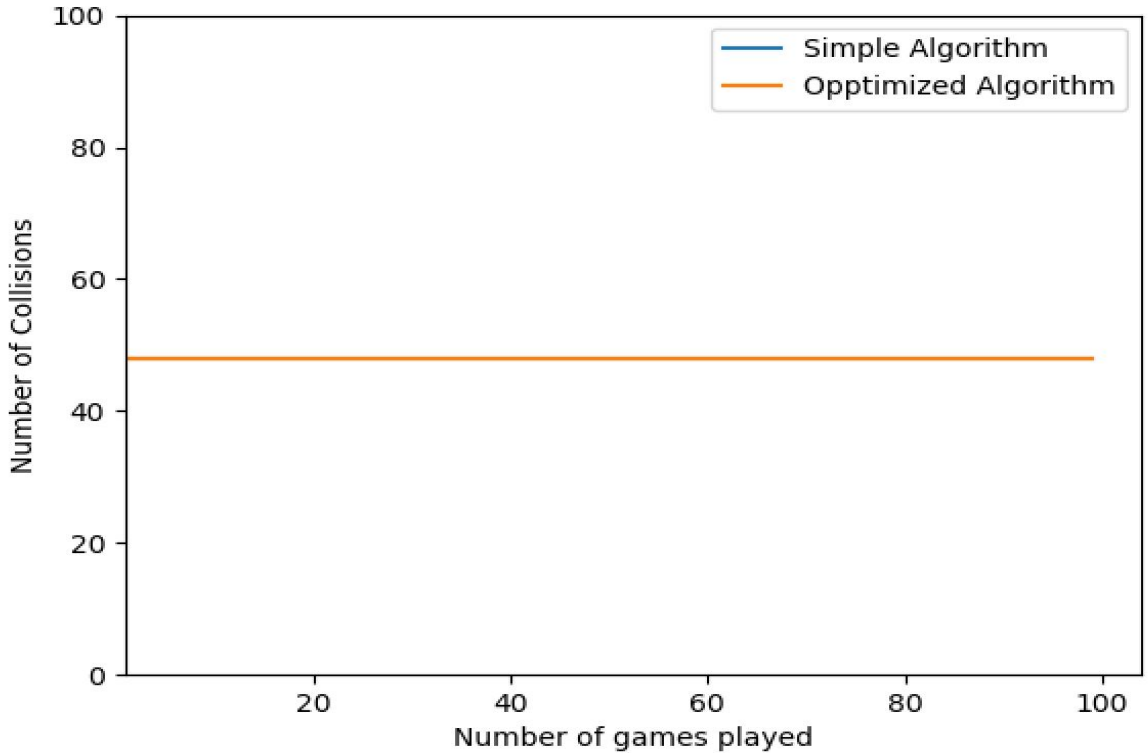

Figure 6:

The parameters used for the testing in Figure 6 are:

M (Number of Tracks) = 2
S (Player Time on Each Track) = 10

Out of the 100 tests conducted, the algorithms perform the exact same.