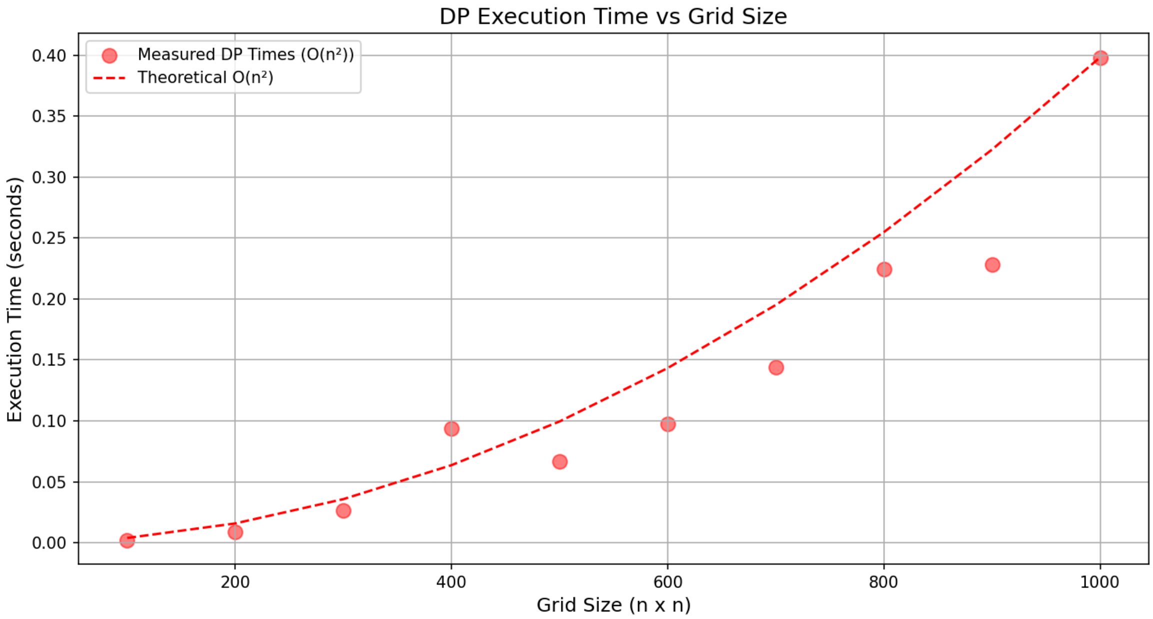
Project 2 Report - Opponent Avoidance Problem

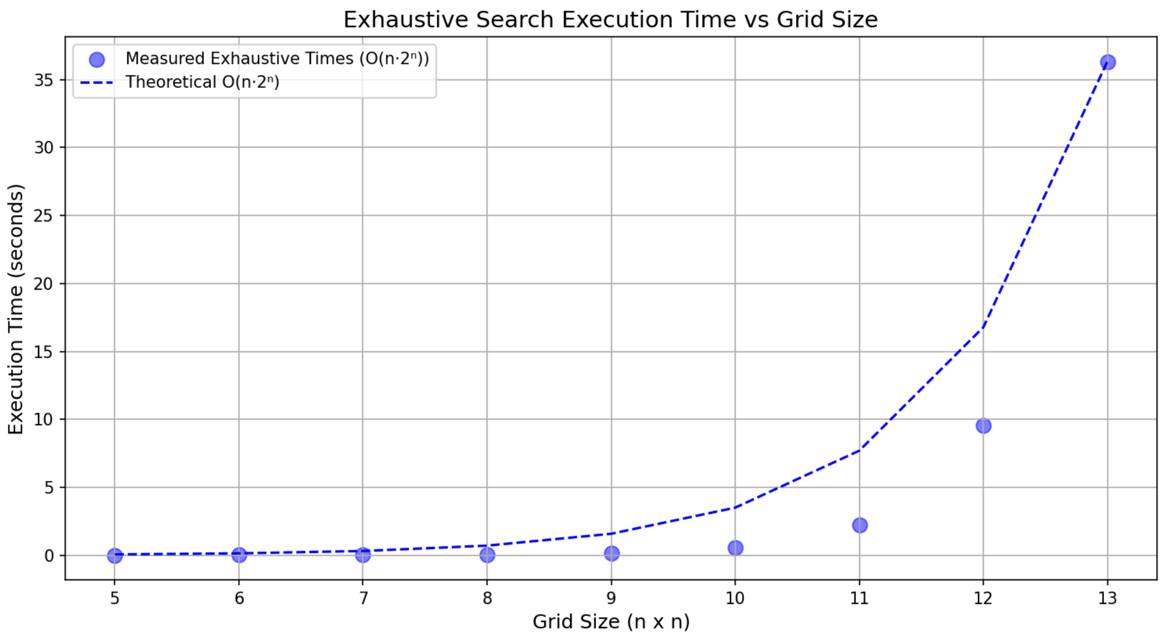
Chanran Kim (shining04@csu.fullerton.edu)
Brian Alvarez (briandalvarez@csu.fullerton.edu)
Christopher Contreras (CContreras71@csu.fullerton.edu)
Vinh Nguyen(vinhgod123@csu.fullerton.edu)

A. Scatter plots for each of two algorithms.

```
DP Sizes: [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000]
DP Times: [0.002339, 0.009051, 0.026383, 0.093923, 0.067018, 0.097275, 0.143976, 0.224502, 0.227996, 0.398178]
Exhaustive Sizes: [5, 6, 7, 8, 9, 10, 11, 12, 13]
Exhaustive Times: [0.001361, 0.002498, 0.008583, 0.035213, 0.11989, 0.540318, 2.244695, 9.523191, 36.345332]
```



Dynamic Programming Execution Time vs Grid Size



Exhaustive Search Execution Time vs Grid Size

Using exhaustive search follows a time complexity of $O(n.\ 2^n)$, while dynamic programming (DP) follows a time complexity of $O(n^2)$, as confirmed through a scatter plot. Although the exhaustive algorithm can obtain all possible paths, it takes too much time to be practical. On the other hand, using the DP algorithm significantly reduces the time required to find the solution.

B. Answers to the questions (Each answer should be at least one complete sentence.)

I. Is there a noticeable difference in the performance of the two algorithms?

There is a noticeable difference in performance as Dynamic programming algorithms perform a lot better when the grid size increases. On the other hand, Exhaustive search peaked at 7 seconds when it was barely 12 grid size.

II. According to your experimental observation, which of the implementation is faster, and by how much?

The implementation that is faster and even more efficient would be the Dynamic programming. Dynamic programming is faster by a huge amount. As you can see in the output and the scatter plots, even at 500 grid size it is still faster than Exhaustive search at 12 grid size by a whole 7 seconds.

III. Are your empirical analyses consistent with the predicted big-O efficiency class for each algorithm? Justify your answer.

For the exhaustive search, our analysis is consistent with the predicted big-O efficiency because as you can see, the ES has an exponential growth which demonstrates a high complexity time faulting in a big-O notation of O(2^n). It is too slow as the grid number move up. As for dynamic programming, it demonstrates a polynomial growth which results in slower time as the grid number moves up. Therefore, the notation of dynamic programming is O(n^2).

IV. Is this evidence consistent or inconsistent with hypothesis 1? Justify your answer.

With the evidence that we've collected, we can support that the hypothesis is consistent. The reason for this is Exhaust search is a O(2^n) notation and therefore implements an exponential running time. Which means that it would take its time to search for all possible solutions resulting in a longer running time. It double checked every solution and eventually resulted in the most optimal solution.

V. Is this evidence consistent or inconsistent with hypothesis 2? Justify your answer

With all the evidence that we've collected, we believed that the hypothesis would be consistent with our evidence. Since, Exhaustive search takes extremely long for any algorithm to use as it is an $O(2^n)$. Therefore, it would hinder any program that tried to use this algorithm for practical use.