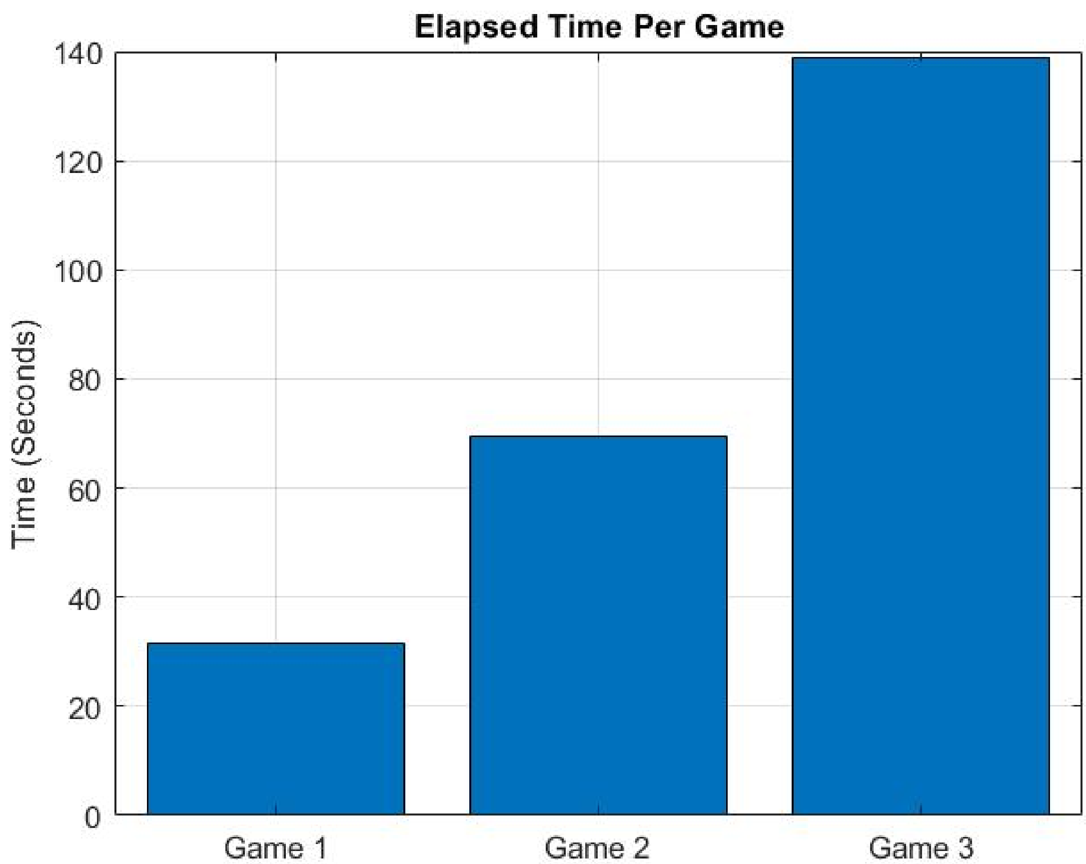
Brooke Albanese

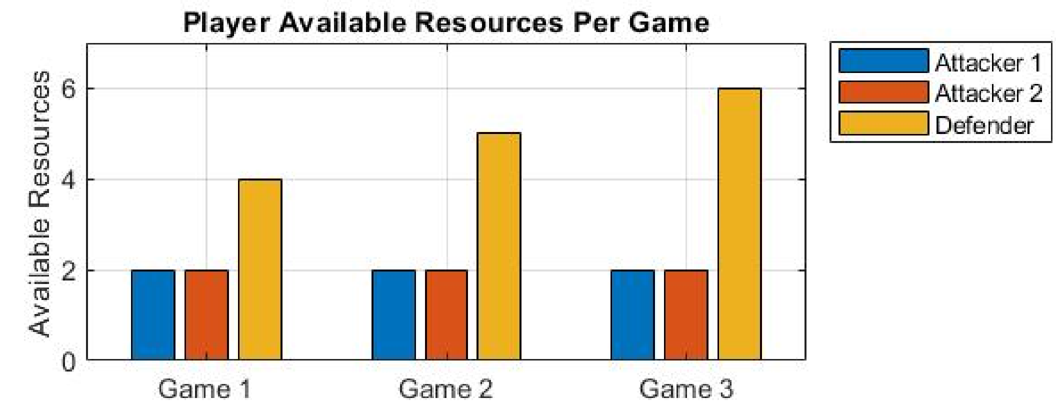
EECE 580A Cyber Physical Systems Security

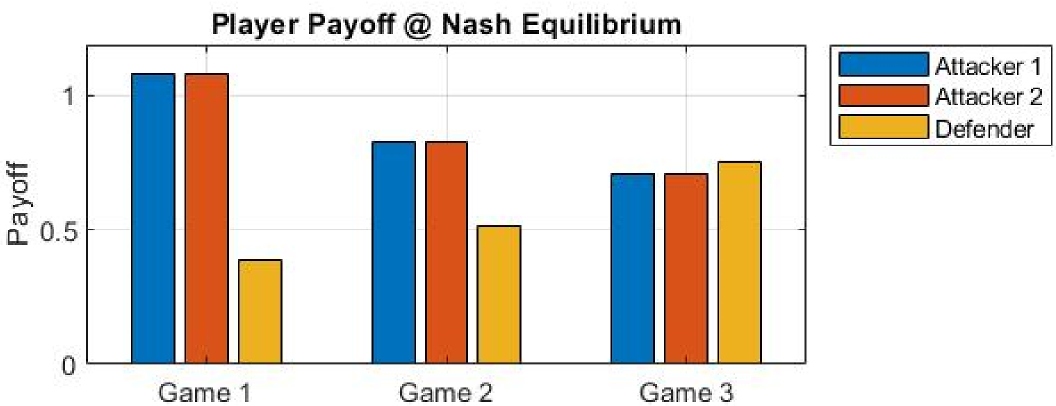
The goal of my project was to determine if the computation time of the Colonel Blotto Game, implemented via MATLAB simulation, could be decreased if we approached the problem differently. In order to compare the computation time of the previous year's project(s) and the new dominating strategy proposal I created four separate folders in the GitHub repository to ensure that the code updates were kept separate.

Folder 1: ***Original without Dominating Strategy Reduction***

This folder includes code that runs the original Colonel Blotto Game simulation, prior to removing the dominated strategies. Running this gives us the following plots to use as a baseline:

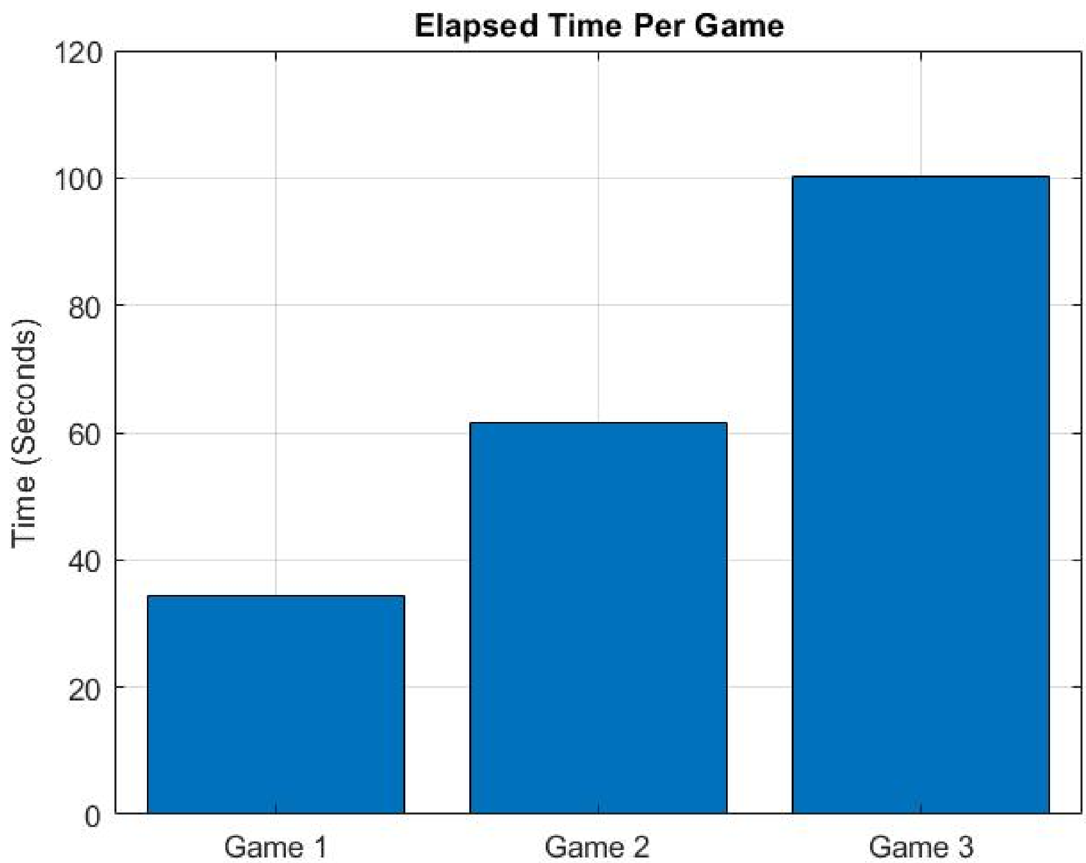


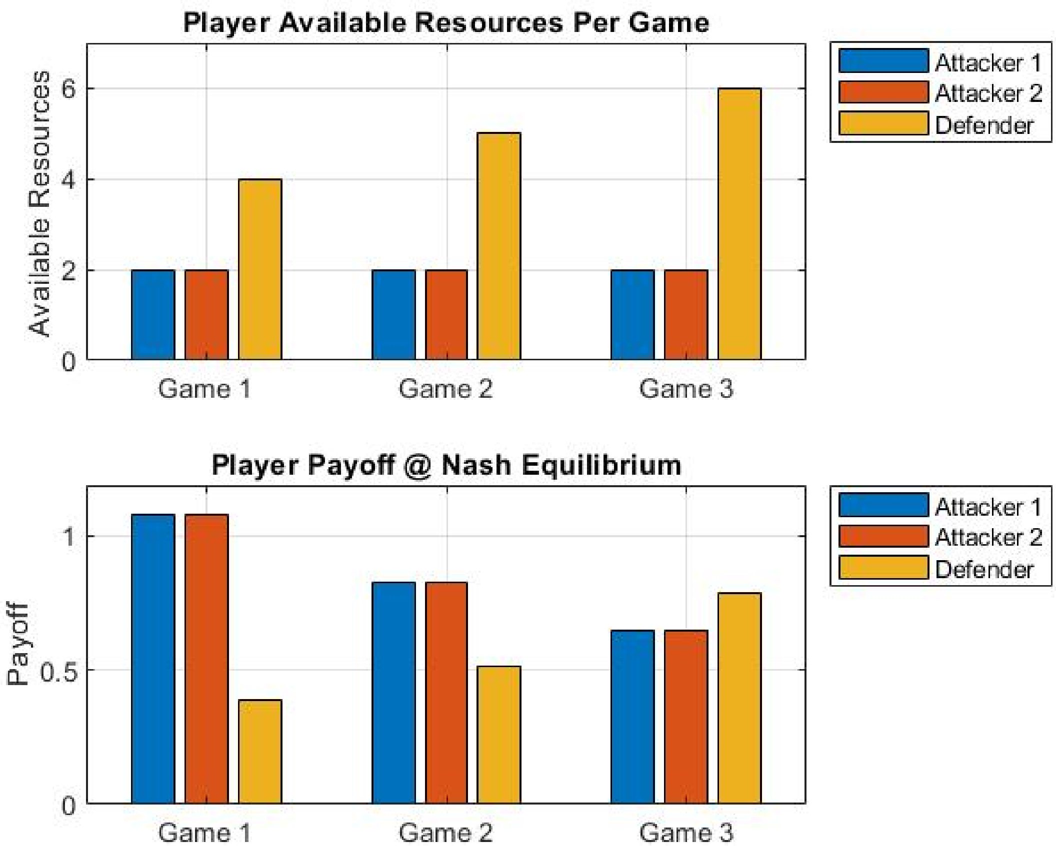




Folder 2: ***Dominating Strategy Reduction (last year's project)***

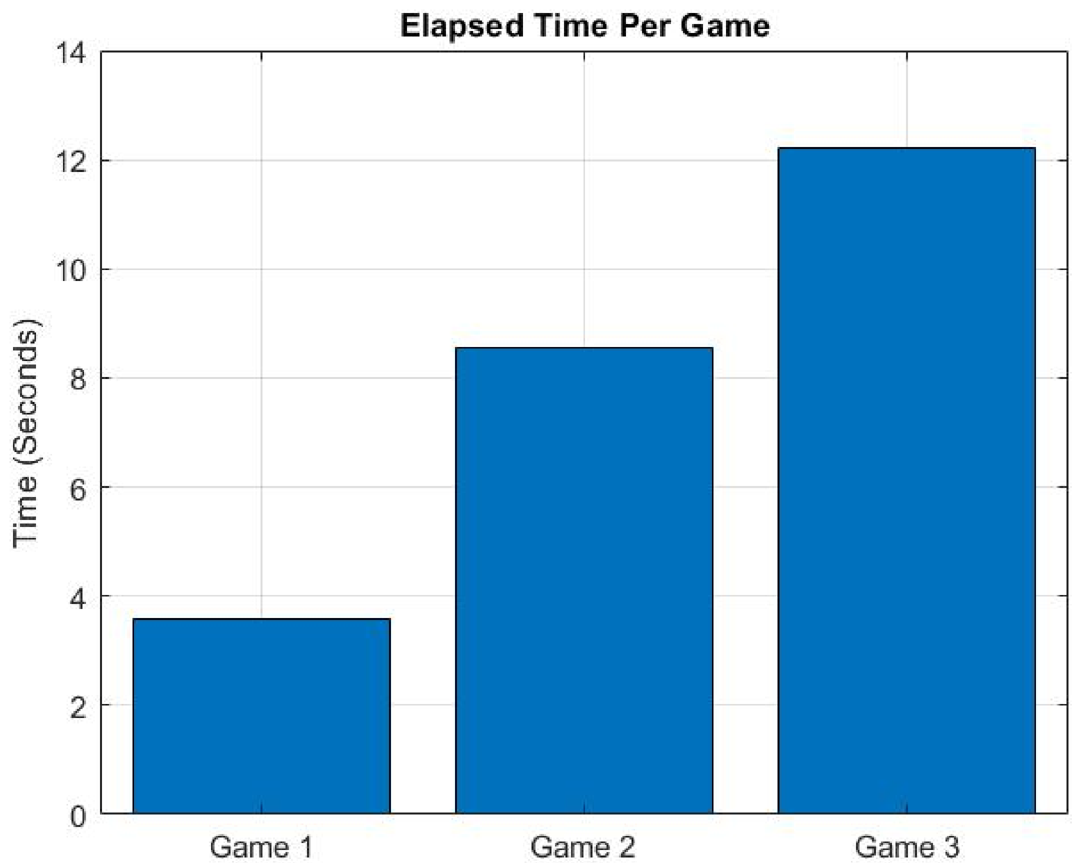
This folder includes code that runs the Colonel Blotto Game simulation with the dominating strategy improvements. Running this gives us the following plots, which show that the dominating strategy improvements resulted in a 40 second decrease in computation time for *Game 3*, which had the longest computation time in the original game.

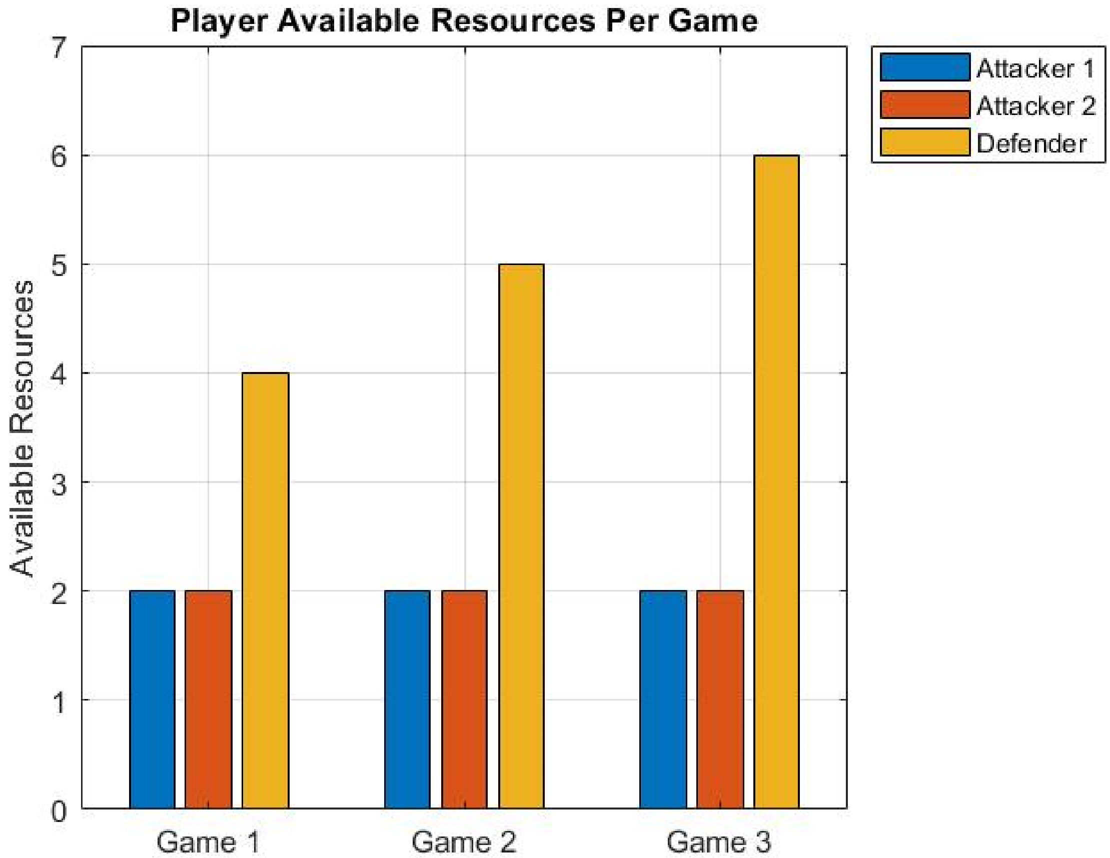

Elapsed Time Per Game
Time (Seconds)
0
20
40
60
80
100
120
Game 1
Game 2
Game 3

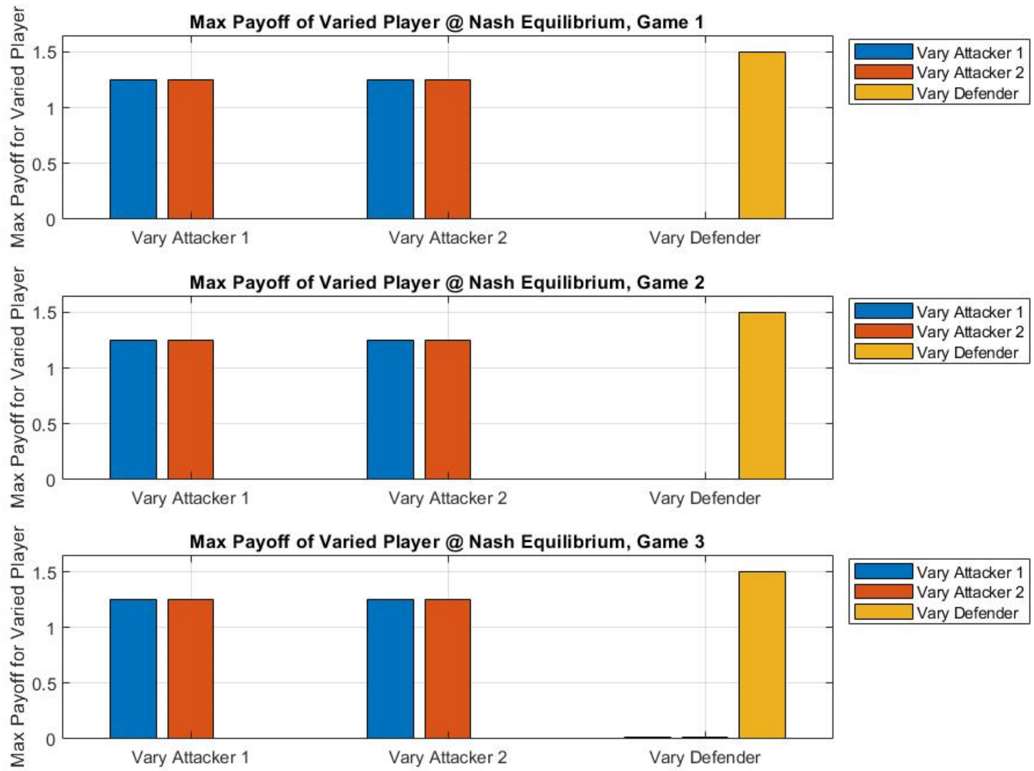


Folder 3: ***Vary One Player's Strategies***

This folder includes code that runs the Colonel Blotto Game simulation with the new dominating strategy proposal. Essentially, the idea here was to hold the strategy of two of the three players' strategies constant and solve for the Nash equilibrium over the entire strategy space of the third player, then update the two player's constant strategies and re-solve for the Nash equilibrium over the entire strategy space of the third player again. Using this algorithm substantially decreases the computation time. However, as you will see in the plots, maximizing one player's payoff while holding the others constant causes differences in the Nash equilibrium compared to the above; this essentially means that in order to properly interpret the results from this new method we will have to find a different way of analyzing them. Following are plots of the game computation time, player resources per game, and a plot of the payoffs corresponding to the maximum payoff of the player whose entire strategy set is being considered for each game:

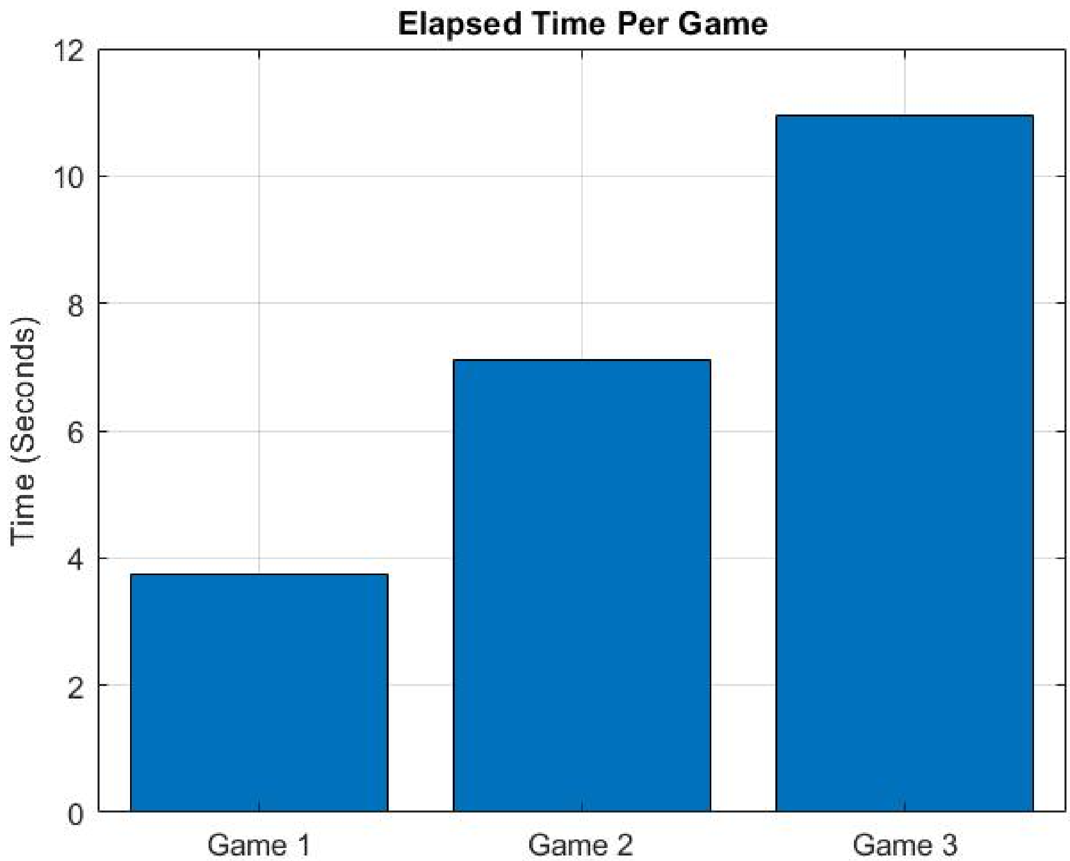

Elapsed Time Per Game
Time (Seconds)
0
2
4
6
8
10
12
14
Game 1
Game 2
Game 3

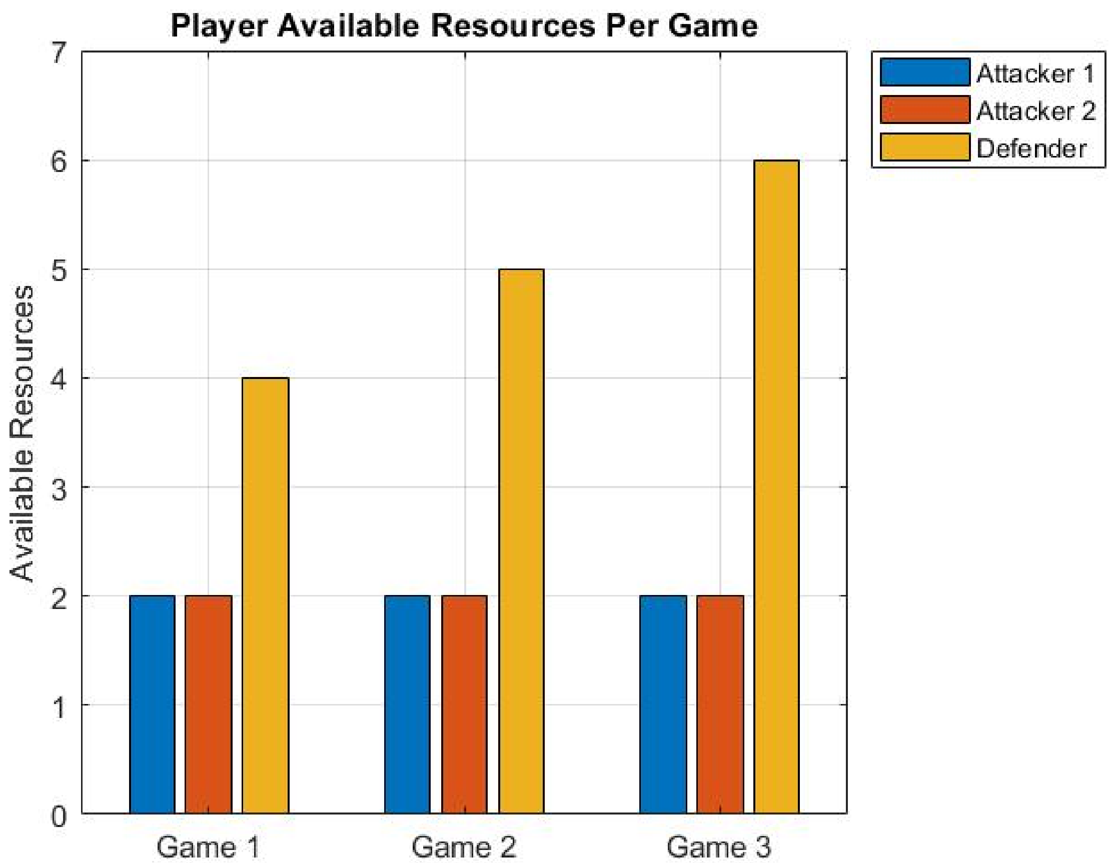

Player Available Resources Per Game
Available Resources
0
1
2
3
4
5
6
7
Game 1
Game 2
Game 3
Attacker 1
Attacker 2
Defender



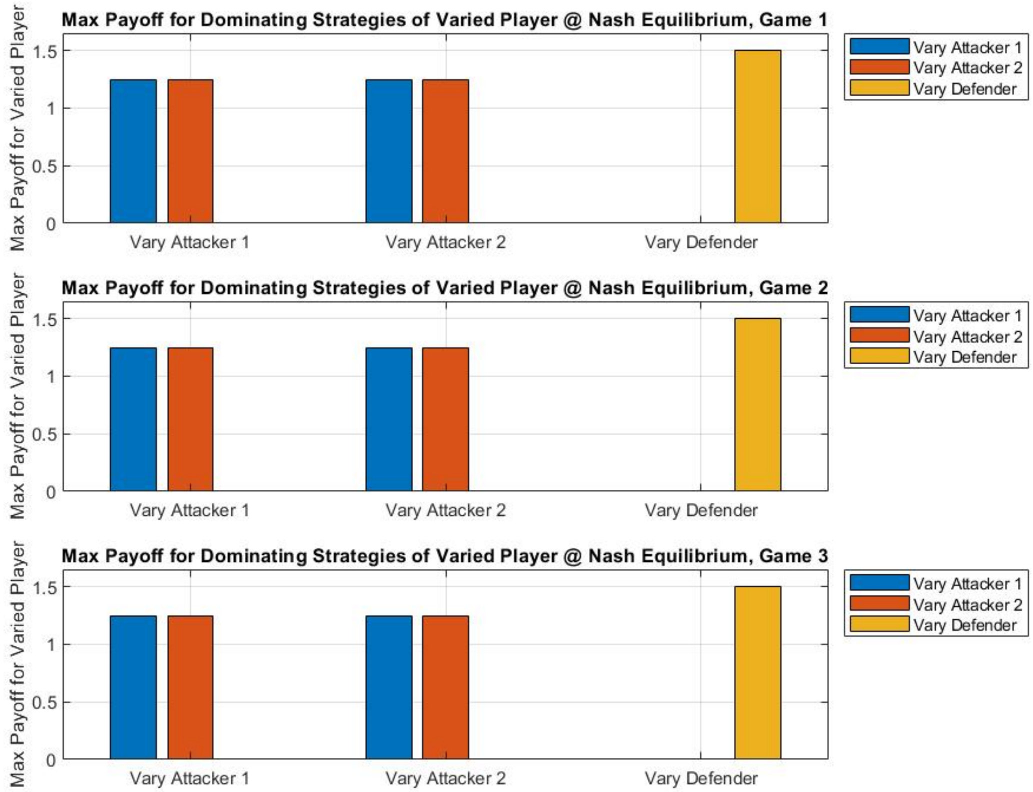
Folder 4: ***Vary One Player's Strategies with Dominating Strategy Reduction***

This folder includes code that runs the Colonel Blotto Game simulation with the new dominating strategy proposal combined with the dominating strategy proposal from last year's project. Following are plots of the game computation time, player resources per game, and a plot of the payoffs corresponding to the maximum payoff of the player whose entire strategy set is being considered for each game:


Elapsed Time Per Game
Time (Seconds)
0
2
4
6
8
10
12
Game 1
Game 2
Game 3



Player Available Resources Per Game
Available Resources
0
1
2
3
4
5
6
7
Game 1
Game 2
Game 3
Attacker 1
Attacker 2
Defender



The code I wrote (Folder 3) and the combination updates I made (Folder 4) both significantly decrease the time it takes the code to run. Future work includes increasing the number of resources and/or nodes, as well as changing the threshold to see if we find similar results with folder 3 and folder 4 code. Other future work could include improving the new algorithm and updating the plots to allow for easier comparison between different algorithms.