Matrix Analysis

Number of Rounds

For our simulation, we first decided on the number of rounds to use. We opted for 100 rounds as this number is sufficiently high for the algorithms to learn and apply their strategies. Additionally, between 50, 100, and 200 rounds, the results for each matrix were very similar.

Defined Agents

We defined several agents to make the simulation more varied:

- AlternatingAgent.java: AlternatingAgent alternates its actions between cooperating and defecting in each round.
- DAgent.java: DAgent is a deterministic agent that always plays D.
- HAgent.java: HAgent is a deterministic agent that always plays H.
- NN_Agent.java: NN_Agent uses a neural network to decide its action based on the history of plays.
- RandomAgent.java: RandomAgent selects its actions randomly in each round.
- RL_Agent.java: RL_Agent uses reinforcement learning to optimize its strategy based on received rewards.
- TFT.java: TFT (Tit for Tat) cooperates in the first round and then mimics the opponent's previous action.

Matrices for Simulation

For the simulation, we used two agents of each type mentioned above and defined several matrices to compare the results with each of them. We used an abbreviated matrix notation to more easily represent the values of each one. The first digit is the penalty for playing H vs. H, the second the benefit of playing H vs. D, the third the benefit of playing D vs. H, and the last the benefit of playing D vs. D.

1. Original Matrix: -1 10 0 5

   Description: This matrix represents the classical Hawk-Dove game. When both players choose Hawk (H), they both receive a penalty (-1). When one player chooses Hawk and the other chooses Dove (D), the Hawk gets a high reward (10) while the Dove gets nothing (0). When both choose Dove, they both get a moderate reward (5).

2. Higher Penalty for Conflict (H, H): -5 8 2 6

Description: This matrix increases the penalty for mutual Hawk (H) actions to discourage aggressive strategies. The reward for Hawk against Dove is slightly reduced (8 for Hawk, 2 for Dove), and cooperation (D, D) is rewarded with 6 points each.

3. Incentive for Cooperation (D, D): -2 10 1 7

   Description: This configuration offers a higher reward for mutual Dove (D, D) actions (7 points each) to encourage cooperation. The Hawk (H) against Dove (D) scenario retains a high reward for Hawk (10) but gives a minimal reward (1) to Dove. The penalty for mutual Hawk actions is moderate (-2).

4. Balanced Risk and Reward: -3 9 2 6

   Description: This matrix aims to balance the risk and reward by giving a significant penalty for mutual Hawk (H) actions (-3). The reward for Hawk against Dove is moderately high (9 for Hawk, 2 for Dove), and mutual Dove actions are rewarded fairly (6 points each).

5. High Reward for Dove Action: -4 7 3 8

   Description: This configuration provides a high reward for mutual Dove (D, D) actions (8 points each) and a decent reward for Dove when facing Hawk (3 points). The penalty for mutual Hawk actions is substantial (-4), while Hawk against Dove still gives a reasonable reward (7).

6. Conflict-Focused Matrix: -10 10 0 5

   Description: This matrix imposes a very high penalty for mutual Hawk (H) actions (-10) to significantly discourage conflict. The reward for Hawk against Dove remains high (10 for Hawk, 0 for Dove), and mutual Dove actions are moderately rewarded (5 points each).

7. Cooperation-Focused Matrix: -2 7 2 9

   Description: This configuration strongly encourages cooperation by giving high rewards for mutual Dove (D, D) actions (9 points each). The penalty for mutual Hawk actions is low (-2), and Hawk against Dove yields moderate rewards (7 for Hawk, 2 for Dove).

8. Intermediate Matrix: -3 6 1 7

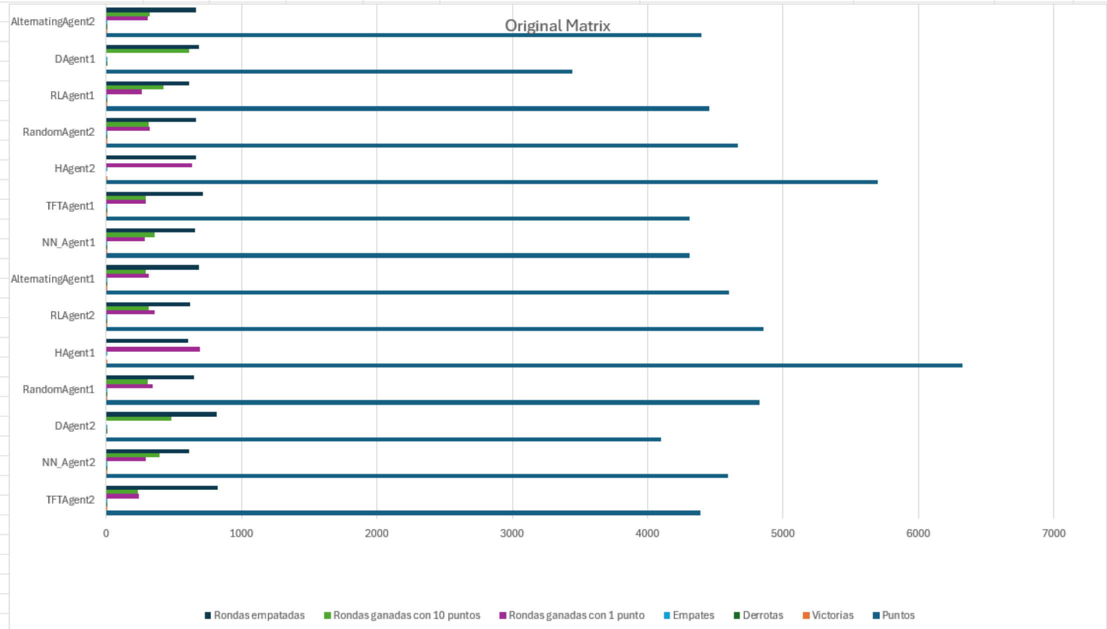
   Description: This matrix offers a middle ground with balanced incentives. The penalty for mutual Hawk (H) actions is moderate (-3), and the reward for Hawk against Dove is modest (6 for Hawk, 1 for Dove). Mutual Dove actions are fairly rewarded (7 points each).

Modifications and Process for Simulations

We ran the simulation with each of the matrices and 100 rounds to compare the results obtained.

We modified the Main Agent to define 100 rounds and also to write the results of each agent to a CSV file at the end of each game (using the statistics created for the GUI in the first version). The CSV file is named with the rounds and matrix parameters, for example: results_100_-2_7_2_9.csv.

Nombre del jugador	Puntos	Victorias	Derrotas	Empates	Rondas ganadas con 1 punto	Rondas ganadas con 10 puntos	Rondas empatadas
TFTAgent2	4393	3	2	8	239	238	823
NN_Agent2	4596	3	8	2	294	394	612
DAgent2	4100	0	11	2	0	480	820
RandomAgent1	4828	7	4	2	343	309	648
HAgent1	6323	10	0	3	693	0	607
RLAgent2	4858	8	4	1	358	318	624
AlternatingAgent1	4602	4	7	2	317	297	686
NN_Agent1	4306	4	4	5	289	357	654
TFTAgent1	4309	3	2	8	291	290	719
HAgent2	5696	12	0	1	636	0	664
RandomAgent2	4664	6	5	2	322	316	662
RLAgent1	4452	4	8	1	264	424	612
DAgent1	3445	0	10	3	0	611	689
AlternatingAgent2	4400	5	4	4	310	322	668

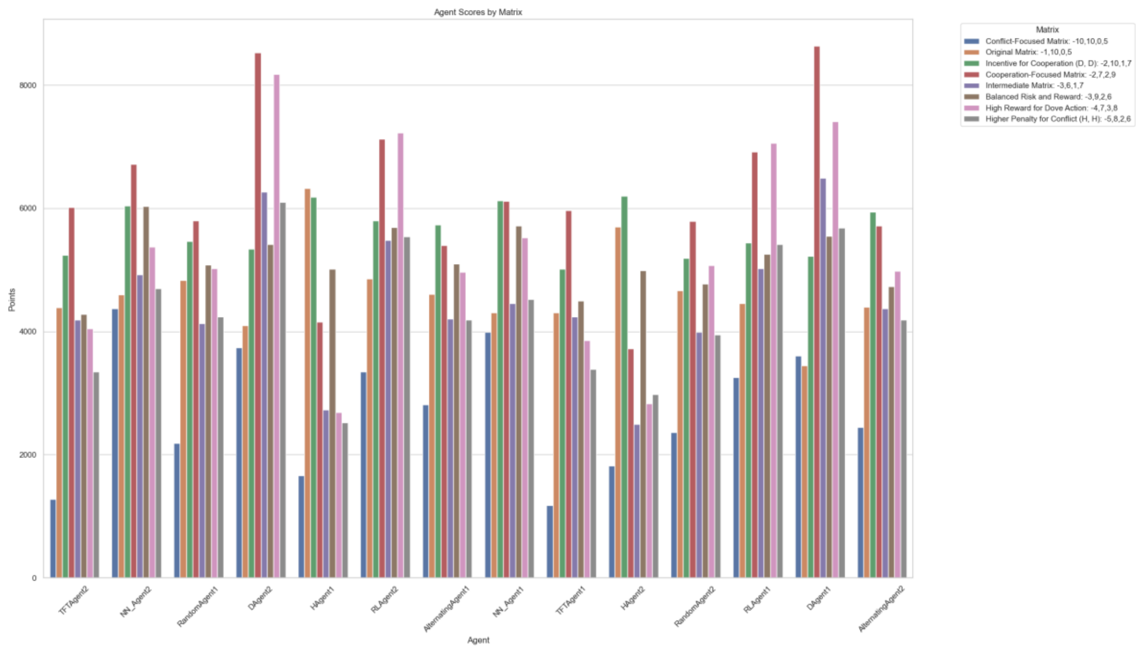


We ran the simulation for the different matrices, obtaining a CSV file for each matrix. Subsequently, we combined the data from these CSV files into one and used another script to create graphs to show the results.

Analysis of Results

In our analysis, we focused only on the final scores (as these determine the victory) and aimed to balance the strategies and especially penalize deterministic agents (always playing H).

The first figure shows the result of each matrix for each agent. It already indicates that deterministic agents (always playing D) obtain significant benefits when the reward for cooperation is increased, meaning these matrices would benefit deterministic agents and would not be suitable.

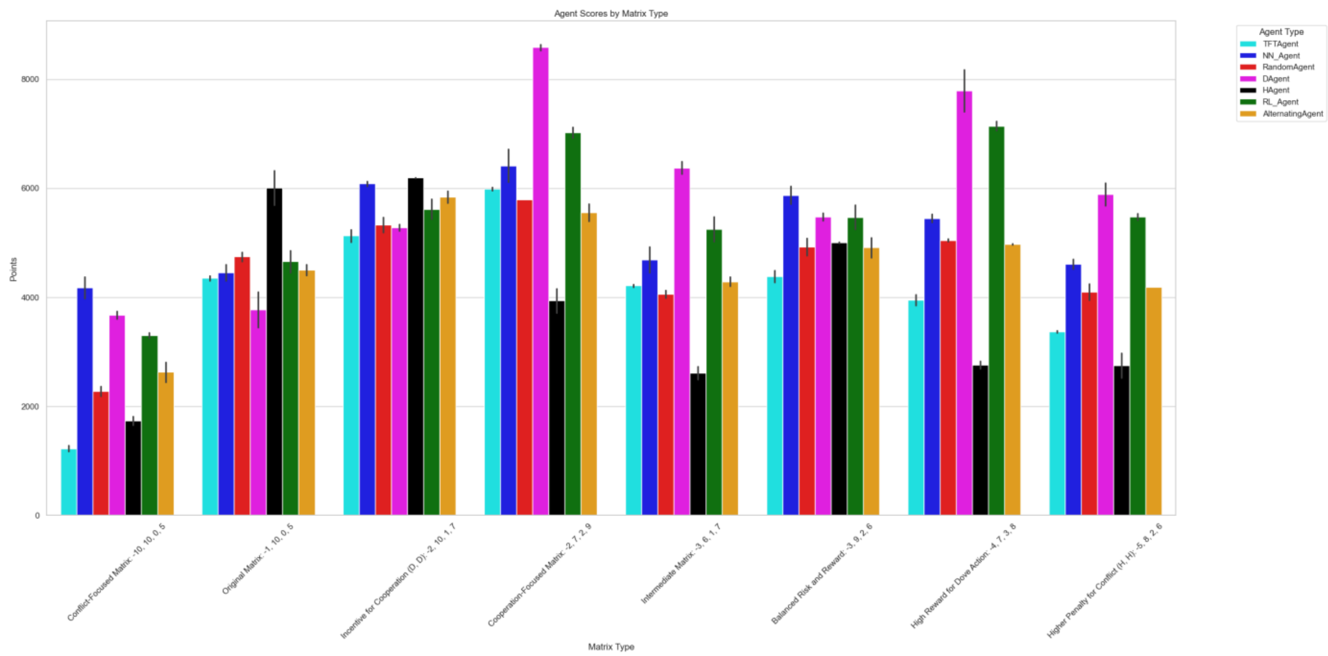


The second figure is the most representative, showing the results of each agent for each matrix type. Agents of the same type are grouped in the same bar, using box plots to mark the differences between the two agents of the same type for a more readable graph. Deterministic agents (always playing H) are marked in black as they are our main problem to address regarding the original matrix.

From left to right, the analysis is as follows:

1. The first matrix heavily penalizes conflict, resulting in lower overall scores compared to other matrices. This is good as the best results are for the NN and RL agents (and also for DAgent since the conflict penalty allows it to gain more points).
2. The original matrix behaves as expected, with HAgent winning by a significant margin.
3. The next matrix, which slightly increases the cooperation reward, yields better results, with the NN agents nearly matching the HAgent, but it is still insufficient.
4. The Cooperation-Focused Matrix seriously penalizes HAgent but overly benefits the deterministic DAgent.
5. The Intermediate Matrix heavily penalizes HAgent compared to the original, but the increased cooperation reward makes DAgent very powerful.

6. The Balanced Risk and Reward Matrix seems the most appropriate, with NN and RL agents obtaining the top positions, and the deterministic H agent being almost on par with other non-AI strategies.
7. The matrix that heavily rewards D (Cooperation-Focused) yields the expected results, heavily benefiting DAgent.
8. The last matrix penalizes conflict and favors cooperation to a lesser extent than the third or fourth matrix, resulting in more favorable results for the agents but unbalancing HAgent against DAgent.



NN and RL agents adapt best and achieve the best results across most matrices. They only dominate in two matrices, but these are among the most complex, and they remain in the top 3 positions across all matrices.

The matrix with values (-3, 9, 2, 6) is the most suitable for a tournament with NN and RL agents as it sufficiently penalizes deterministic strategies to prevent them from winning, and it also ensures that other strategies like Random, Tit for Tat, or Alternating have worse results. This would result in a more interesting tournament where agents need to develop more diverse strategies.