The influence of an Agent's initial aggressiveness on the Hawk-Dove equilibrium

Olaf Boekholt (4609298), Silas Peters (4419197), Fabian Rutten (1566393)

April 7, 2023

Abstract

The Hawk-Dove game is a classic model in game theory that explores the dynamics of competition and cooperation in social situations. In this game, two players can choose to either behave aggressively like a hawk or passively like a dove, in a contest for a limited resource. In our research, we will explore whether there is a correlation between a player's initial aggressiveness and the number of iterations it takes in a computer simulation before both players' aggressiveness form an equilibrium. We explain that we expect that there is a positive correlation between a player's initial aggressiveness and the number of iterations after which the Hawk-Dove equilibrium takes the form (the mean equilibrium iteration).

To test this, we set up a computer simulation using .NET 7, in which we repeat an algorithm for every possible combination of both players' initial aggression rates. The algorithm simulates a single Hawk-Dove scenario a hundred times and calculates the mean equilibrium iteration every time, after which we store the mean of all calculations thus belonging to a single combination. The program leaves us with a spreadsheet in which for all possible combinations of initial aggressiveness the mean equilibrium iteration is given.

Results show a steady growth in initial mean equilibrium iteration when a player's initial aggressiveness grows. Thus, we concluded that there indeed is a correlation between a player's initial aggressiveness and the number of iterations it takes in a computer simulation before both players form an equilibrium in aggressiveness.

Keywords: Hawk-Dove, Conflict, Aggressiveness, Equilibrium

1 Introduction

1.1 The game

The Hawk-Dove game theory studies the behaviour of two players, or *Agents*, in contention for the same resource. An Agent can either choose to be aggressive and try to obtain as much of the resource as possible, i.e. the "Hawk" strategy, or they can choose to be passive and accept sharing the resource, the "Dove" strategy. Both Agents first choose their strategy, and then based on that choice, gain a reward. If the Agents choose a different strategy compared to each other, meaning one Agent chooses Hawk and the other Agent chooses Dove, the Agent that acts aggressively gets the entire reward, and the other gains nothing. If both choose Dove, the two Agents share the reward. The most interesting case is when both Agents choose to be Hawk. The ensuing conflict causes the value of the resource to go down. This gives rise to the following payoff matrix, as shown in the table below:

	hawk	dove
hawk	(V-C)/2, (V-C)/2	V, 0
dove	0, V	V/2, V/2

Hawk-Dove Payoff Matrix

Here, V represents the value of the resource, and C represents the cost of the conflict. In the Hawk-Dove scenario, The cost of conflict C is always higher than the value of resource V, meaning that if both players are aggressive, they get a negative reward, i.e. they get a punishment. This means the Agents have to be very careful and it actively discourages overly aggressive behaviour.

1.2 Other games

There are also a few very similar games studied by game theory, all consisting of a conflict between two players and the option for each player to be either passive or aggressive. Hawk-Dove is a variant of the chicken game. There is a famous analogy for the chicken game, where two cars are driving towards each other but between them, there is a one-lane bridge. The players can either choose to go straight or swerve out of the way. If both choose to be aggressive, the cars crash into each other, representing a very high punishment. If either one swerves, the other player has a clear passage and gets rewarded. If both players swerve, nothing happens.

Most similar to Hawk-Dove is the Prisoner's dilemma. Here the analogy states two prisoners that are about to be sentenced to prison can choose to betray the other and raise their sentence by informing on the crimes they committed, in return for a reduced sentence for themselves. If both choose to betray the other prisoner, both prisoners only get a small amount of time added to their sentence. The payoff matrix is exactly the same as for Hawk-Dove, but now the cost of the conflict is always less than the value of the reward, and two aggressive players still get a positive reward. This encourages both players to act aggressively.

The last variant is called the Snowdrift dilemma. Two cars want to pass a road that is completely covered in a snowdrift. Players can choose to shovel away the snow, analogous to being passive, or to stay in their car and let the other do all the work, the aggressive strategy. Two passive players obtain the highest reward because then the snow is cleared the fastest. In this game, being passive always leads to a positive reward, because eventually the snow gets cleared and the players can continue along the road.

1.3 Motivation

The principles of these three games are all very similar to those of the Hawk-Dove game theory, the main difference is how the payoff matrix is filled in. For this research, we chose to study the Hawk-Dove variant, because it seemed the most interesting to us. As mentioned before, being too aggressive leads to punishment, but being too passive might result in you never gaining anything at all, which means the choice of what strategy to use can be quite tough. There is not a single strategy that gets actively encouraged all the time, unlike the Prisoners dilemma, where being aggressive always leads to a positive outcome for yourself, or the Snowdrift dilemma, where being passive always ensures a reward.

There are also many real-life applications of the Hawk-Dove game theory, which we will elaborate more on later in section 2.1. For this research, we will be simulating iterations of conflicts, and studying the ensuing behaviour of the two Agents playing a game of chance, based on their initial level of aggression.

2 Background

The first version of the Hawk-Dove game theory was described in 1973 in the paper "The Logic of Animal Conflict" by Maynard Smith and Price [MP73]. Since then, there has been a lot of research focused on the biological and evolutionary aspects of the theory. Auger, Bravo de la Parra, and Sánchez studied the long-term effects of Hawk-Dove conflicts on the coexistence and stability of the community of the two populations involved in the game [AdS98]. There has also been some previous research about the equilibria that occur in Hawk-Dove [dPK22]. Blazquez de Paz and Koptyug state: "The Hawk-Dove game has two possible Nash equilibria in pure strategies - one of the players behaves as a hawk and the other as a dove". A Nash equilibrium is an equilibrium where a player can achieve the highest possible reward for themselves by never deviating from their initial strategy. Each player's strategy is optimal when considering the decisions of other players, meaning all outcomes of future conflicts will remain the same. The (Hawk, Dove) and (Dove, Hawk) equilibria naturally follow the rules of the game, because if either player deviates from their strategy, their own score will go down, as can be seen in the payoff matrix shown in 1.1.

2.1 Real life applications

Hawk-Dove game theory has been used to model real-life conflicts. Most notably by Li et al. in 2022 [LHQA22], who used it to analyse the China-Australia iron ore trade conflict. China imports more than 60% of its iron ore from Australia but is now looking at other countries. Australia exports 80% of its iron ore to China, meaning this trade has a major influence on the economy of both nations. However, there have been numerous incidents between the companies of the two countries, putting this trade under pressure. Li et al. construct the RDEU hawk-dove game model of the iron ore trade conflict between Australian iron ore companies and Chinese steel companies. There are also examples of real-life populations of a bird species in Australia following the Hawk-Dove principles of aggressiveness when reproducing, as shown by Kokko, Griffith, and Pryke [KGP14]. More aggressive red and more passive black Gouldian finches need each other to ensure proper fitness for their offspring and are therefore locked in an eternal battle for food and shelter. Too many offspring of one type could prove to be catastrophic for the entire population.

Thus, we can see that there has been some research about the Hawk-Dove game theory, mainly focused on the outcomes of the various mentioned. However, with this research, we want to focus on a different aspect of the equilibria, not mentioned in any previous research, namely how fast they occur.

3 Research question

Following our introduction, the Hawk-Dove principle can thus be applied to many scenarios and forms a basis for future research. There is, however, one aspect of this principle that might influence all Hawk-Dove scenarios: the occurrence of an equilibrium between both Agents' chance of acting as a Hawk - Hawk-Dove equilibrium, based on their initial aggression. That is, both Agents undergo many confrontations with each other, such that both Agents learn from each other. Knowing how fast the Hawk-Dove scenario will lead to equilibrium can be important in predicting the outcome of Hawk-Dove scenarios and might simplify future research. It will be relevant to all Hawk-Dove scenarios. To gain insights into when a Hawk-Dove equilibrium might occur, we will try to answer the following research question: *"How does an Agent's initial aggressiveness in a hawk-dove scenario influence the eventual aggressiveness-equilibrium between two Agents, if one occurs?"*

4 Hypotheses

4.1 The hypotheses

To answer our research question, we will focus on a more quantifiable aspect of said equilibrium: how early the equilibrium starts to form. First, we will define the status quo (H_0) and our alternative hypothesis (H_A).

- H_0 = There is no positive correlation between an Agent's initial aggression and the iteration at which the equilibrium occurs.
- H_A = There is a positive correlation between an Agent's initial aggression and the iteration at which the equilibrium occurs.

4.2 Reasoning behind the hypotheses

Our hypotheses are based on the theory that Hawk-Dove scenarios form a Nash equilibrium [dPK22]. The following is a more detailed reasoning which supports our hypotheses and said theory:

We first define X_h, which is the aggressiveness and hence the chance of acting as a hawk of Agent X. X_d is its inverse: the chance of acting as a dove.

We believe that if an Agent A undergoes many confrontations and his initial aggression is higher than that of the other Agent B, then Agent B will back off more than A because B has a higher chance of acting as a dove and thus rewarding A to keep playing as a hawk.

If we define A to be aggressive, that means that its chance of acting as a hawk A_h is higher than its chance of acting as a dove A_d. Thus, $A_h > A_d$ and $A_h > 0.5$. Given an arbitrary aggression rate B_h, if $A_h > 0.5$ then the chance that $B_h \leq A_h$ is more than 0.5. Hence, in general, if Agent A is aggressive then the chances are bigger that Agent B is less aggressive than more aggressive, and we can expect Agent B to yield more often than Agent A. This causes Agent A to act as a hawk more and more often, eventually leading to an equilibrium where Agent A always acts as a hawk.

In short, it is assumed that the chance of B_d becomes higher when A_h becomes higher. Thus, when A_h is initially higher, the time or number of confrontations it will take for B_d to approach 1 is lower. Thus, an equilibrium is assumed to occur earlier when A_h is initially higher.

5 Method

In order to determine which hypothesis we would need to reject, we set up an experiment using a simple computer simulation. Every conflict is simulated sequentially between Agents A and B. Agents A and B are identical in behaviour. Every iteration is one conflict, simulated based on the history of previous conflicts. In every conflict, an Agent decides if it becomes more or less aggressive, changing its chance to become a Hawk in this iteration. Each game is a game of chance. Therefore the iteration where the equilibrium establishes is not consistent, but rather random. That is why our results are a mean over multiple games of chance, each with possibly other results. In our method, we introduce variables that are locked and variables that change in order to answer our research question. The variables that are locked for this given research, as testing those variables would expand the research beyond our scope.

5.1 Agent behaviour

Both Agents have the same behaviour and react with the same set of rules as the previous iterations. When an Agent needs to choose its stance, it is asked what its aggressiveness is currently. When asked, the Agent re-evaluates its aggressiveness based on all the previous iterations. The Agent takes the average of the scores of itself and its opponent. If the average of the opponent is higher, the Agent increases its aggressiveness. Otherwise, the aggressiveness decreases, since the Agent searches for the lowest necessary risk, i.e. aggressiveness, with the highest yield in scores. Each increase or decrease in aggressiveness is with one, with a minimum value of zero and a maximum value of one hundred. The Agent has a locked variable, namely how far back it can look in the history of the iterations. We set the said history to thirty iterations, which means that the Agent takes the score averages of the last thirty iterations. Altering the history decreases the period for each oscillation and makes an equilibrium established at marginally different iterations, at the cost of entropy and accuracy 6.1.

5.2 Our setup

To calculate the thousands of iterations, the games are simulated in software written in C-Sharp .NET 7 [PRB23]. We divided each section of our program into different classes, each taking care of their own responsibilities and calculations for intermediate results, with the main program being responsible for starting, wrapping up and outputting the results. The simulation takes into account that this is a game of chance and that it should be random. Therefore we use the Microsoft C-Sharp Random class. This class is up to the IEEE standard for True Random Number Generation [JBZ20]. Each simulation is deterministic by making sure the seed, the current time of the machine where the simulation is running, is mentioned in the results.

5.3 Main program

The main program does a test for each combination of initial aggressiveness per Agent. To overcome floating point errors [WIL60], we use integers to represent the chance. Where one would normally note '0.6' for a 60% chance, our program states '60'. Tests go as deep as 1%, which means that we test the combination of two Agents with aggressiveness from 0% to 100%. An Agent's behaviour is deterministic given its aggressiveness, which results in the conclusion that a simulation of a game with Agent A with aggressiveness X and

Agent B with aggressiveness Y, has the same result as Agent A with aggressiveness Y and Agent B with aggressiveness X. The simulation is symmetric. Using this logic, the performance constraint for the number of tests is almost cut in half. Using the Cartesian product for every possible outcome, you would have ten thousand outcomes. Using the aforementioned combinations given by a binomial coefficient results in almost half the calculations, still covering all the necessary results.

$$\binom{n}{k} = {}^{n}C_k = \frac{n!}{k!(n-k)!}$$

$$n = 100$$

$$k = 2$$

The binomial coefficient excludes exact matches, but those are included in our research. That is where Agent A has an aggressiveness X and Agent B has an aggressiveness X. Thus, this gives us a total of 5051 tests:

$$\text{exact matches} = 101$$

$$\text{combination} = 4950$$

$$\text{total tests} = 5051$$

In order to have each test run independently from the others and to avoid depleting the seed for the random class, this main program has a so-called *master random* that creates seeds for a separate random for each test. The main program then runs all the tests and collects all the results and necessary data. Since the tests are independent, it is possible to implement this algorithm on a parallel platform, though it is now purely sequential in order to maintain simplicity and focus on the algorithm itself.

5.4 Outer loop

The outer loop is responsible for running enough tests to return an accurate mean. The outer loop has a locked variable for the number of times we run a test with a different random, to take the mean. Keeping performance and accuracy in mind, the amount of tests is set to a hundred independent tests, which proved to be sufficient for results without needing heaps of computational power. In Appendix A is shown that the result drastically changes when taking only one sample. Again, this loop has a *master random*, that creates seeds for new randoms for each test. Each test again is independent and this model can be applied to a parallel model. The outer loop runs each test sequentially and saves the results. It then calculates the average number of iterations it took for the Agent to both reach an aggressiveness of 0 and continue to keep this for the remaining duration of the test. This is the defined equilibrium.

5.5 Inner loop

The inner loop is responsible for a single game of chance between two Agents, a single test. Here more locked variables are introduced and set. The number of iterations is set to a hundred thousand since testing showed that there were no significant outliers beyond this iteration 6.1. This is mostly a performance ceiling, as more iterations are difficult to prove using this current version of the software. The ratio between conflict costs and the resource value. These are set to two hundred and one hundred respectively. The ratio between these two variables changes the amplitude of the oscillation 6.1. Each iteration loop calculates the score of one conflict between the two Agents. Each Agent is asked to give its stance, either Hawk or Dove. First Agent A is asked to give its stance, which it then calculates by first re-evaluating its aggressiveness 5.1. Then Agent A takes a random number from zero to one hundred, representing a chance, from the random class. If that chance is smaller than the aggressiveness from Agent A, then Agent A takes the stance of Hawk, else it takes the stance of Dove.

5.6 Output format

The software takes the desired results and writes them to a *.csv* file format. This file format is compatible with Microsoft Excel, which was primarily used to process the large amount of data that the software generated, and is easily generated in software. The software first writes the locked variables to the output and the seed used to create the aforementioned *master random*, in order to achieve deterministic results. The actual data points are then written in three columns; Column A for Agent A, Column B for Agent B and Column C for the results of the tests between the two Agents. Microsoft Excel proved to be adequate to display the intermediate results in graphs. This is because there was no direct relation between all three variables, but rather separately Agent A and Agent B, giving meaningful information by seeing both of their plots together in the same graph 6.1. For the final graph, Microsoft Excel was not adequate anymore, as we needed to plot the relation of both Agents with their respective results. We used a web application to turn out data in a readable graph [Plo]. The graph is based on Agent A en Agent B as X and Y axes respectively, the result is represented in the colour of each plot in the graph of Appendix B.

6 Results

To test for such called equilibrium, the software first simulated the inner loop and then the outer, resulting in a graph that gives insight as to how any of these equilibria exist and how they are propagated. These intermediate results are the tools to redefine locked variables and fix unnecessary performance constraints. The final results are then a combined algorithm for both loops that is run for every combination of each Agent and a relative initial aggressiveness.

6.1 Intermediate results

In Appendix C is the graph shown for a single game of chance between the two Agents. Agent A and B both start with different initial aggressiveness. The history range, how far back an Agent can evaluate the scores, is the locked variable value of thirty. This illustrative example shows that there occur two oscillations. One is at a very high rate between the two levels of aggressiveness of both Agents, and the other one is the average height where the oscillations take place. The is no explanation for the second oscillation, which would require further testing and research. It is however possible that it is up to chance since the game is simulated for *true random*[JBZ20]. Eventually, the oscillation stops and both Agents discover the benefit of continuously choosing the stance of Dove. The equilibrium establishes around *3450*.
In Appendix D is again a graph shown for the same game. This time the history range was altered for researching the optimal value for the locked variable. Altering the history range to ten instead of thirty means that each iteration has a larger impact on the average that each Agent takes from the previous scores; 300% more impact. This results in the oscillation period dramatically decreasing, with a down words trend towards the equilibrium, resulting in a significantly earlier equilibrium.
Appendix E is no different from the previous two graphs, other than that the history range is altered again, but this time to one hundred, much higher. This makes every iteration not weigh as much on the average scores; 1000% less. This results in a more stable oscillation with a larger period. The oscillation is thus stable that the equilibrium occurs hypothetically outside our scope of a hundred thousand iterations. The software is currently limited to only a hundred thousand iterations. For further testing, the software should be improved and specialized for parallel hardware for efficiency.
Testing for an adequate value for the history range was done using multiple samples of these games, only changing the history range. Note that Appendix C, Appendix D and Appendix E have the same seed, in order to exclude errors by random chance. By trial and error with samples, the history range of thirty allows entropy in the game for longer and does not demand too much performance, all while having readable data.

6.2 Final results

6.2.1 Notes

The graph of the final result can be found in Appendix B. Before examining the results, two things should be clarified:

- It can be seen that 151 dots are missing from the graph. This is because the software we used to generate a coloured scatter plot [Plo] only allows a maximum of 5000 rows as input, whilst our program exported 5151 rows. Hence, some of the dots belonging to Agent B's initial aggression of 0 and 1 are missing.

  The rows which are not visualized contain a mean equilibrium iteration which would be represented with yellowish colours, thus fitting with the pattern emerging in the graph.

  To see the missing rows, see Appendix F.

- There are no dots below the diagonal between the origin (0,0) and the top-right corner (100,100). Essentially, this is because our simulation iterated over $\binom{\{0,\ldots,100\}}{2} \cup \{(x,x) : x \in \{0,\ldots 100\}\}$, not $\{0,\ldots,100\} \times \{0,\ldots,100\}$. Point (40, 20) represents the exact same scenario as point (20, 40); we do not differentiate between Agents A and B - we only look at the combination of the two. Both (20, 40) and (40, 20) should always yield a similar (symmetric) result.

6.2.2 Observations

Regarding the results, it can be seen that near the origin, the mean equilibrium iteration is relatively low compared to other places in the graph. Following the diagonal originating in (0,0), the graph darkens gradually. Hence, the mean equilibrium iteration increases when an Agent's initial aggression increases.

The mean equilibrium iteration ranges between 0 and 5193.95. Except for the values above 4000 are all colours roughly equally present in the graph. There are only a few darker spots with a mean equilibrium iteration above 5000.

6.3 Statistics

We believe that our results do appropriately represent naturally occurring mean equilibrium iterations if it were that our model was fully correct. In other words, even though our code relies on randomness, the output of the program is correctly representing all possible outcomes of every scenario. For a given initial aggressiveness combination of two Agents, the inner loop of our algorithm is repeated a hundred times, and only the mean of those runs is returned as the mean equilibrium iteration of the given combination. Thus, any probability of intermediate outliers is flatted out. This is greatly visualised by the difference in Appendix A and Appendix B. In Appendix A, the inner loop runs only once. In Appendix B, the inner loop is repeated a hundred times.

Now, we apply some statistical formulas on all mean equilibrium iterations to analyse them in more detail.

6.3.1 Standard Error

- $\bar{x} = \frac{1}{n} * \sum_{i=1}^{n} x_i = \frac{1}{5151} * 19,415,935.41 = 3769.35$
- $s^2 = \frac{1}{n-1} * \sum_{i=1}^{n} (x_i - \bar{x})^2 = \frac{1}{5150} * 931,156,000.80 = 180,806.99$

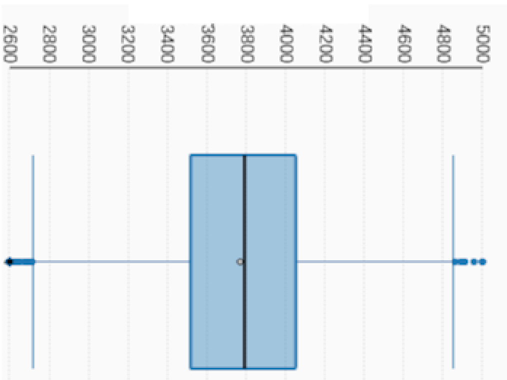
- $s = \sqrt{s^2} = 425.21$
- $SE = \sqrt{\frac{s^2}{n}} = 5.92$

6.3.2 Calculations related to the outliers

- $Q1 = 3516.89$
- $Q3 = 4052.12$
- $IQR = 4052.27 - 3516.97 = 508.23$
- Lower whisker:
  $Q1 - 1.5 * IQR = 3516.97 - 1.5 * 508.23 = 2754.63$
- Upper whisker:

  $Q3 + 1.5 * IQR = 4052.27 + 1.5 * 508.23 = 4814.62$

6.3.3 The outliers



- Lower outliers: 18
  { 0, 425.73, 872.66, 961.34, 1302.11, 1334.84, 1409.78, 1572.84, 1656.66, 1697.97, 1750.14, 1819.86, 1927.51, 1929.64, 2004.44, 2020.87, 2167.44, 2168.43 }
- Upper outliers: 9
  { 5193.95, 5163.65, 5122.36, 5019.75, 4957.05, 4910.48, 4905.85, 4888.52, 4861.13 }

7 Conclusion

Now we try to answer our research question: *"How does an Agent's initial aggressiveness in a hawk-dove scenario influence the eventual aggressiveness-equilibrium between two Agents if one occurs?"*.

Near the origin of the graph in Appendix B, the mean equilibrium iterations are quite low. However, as both Agents' initial aggressiveness increases, the mean equilibrium iterations start to grow. This growth seems to be linear, but that is not supported by any calculation. More importantly, however, even when only one Agent's aggressiveness increases, there will be a growth in mean equilibrium iterations. Hence, it can be said that an Agent's initial aggressiveness in a hawk-dove scenario influences the eventual aggressiveness-equilibrium between two Agents: the more aggressive an Agent is initially, the longer it takes for this equilibrium to occur. There is a positive correlation between an Agent's aggressiveness stance and the mean equilibrium iteration.

As stated in the hypotheses section, we proposed as our alternative hypothesis that there would be a positive correlation. This means that we can accept this hypothesis, and reject the null hypothesis, which stated that there would not be a positive correlation between the initial aggression and the aggressiveness equilibrium.

Our hypotheses were based on the belief that a Nash equilibrium would occur at some point, as suggested by research done by Blazquez de Paz et al. [dPK22] However, as our intermediate and final results show, this is not the case for our model. In all of our simulations, both Agents' aggressiveness gradually shrinks towards zero, and as soon as both hit zero both will stay at zero. This means that the equilibrium our model measures are the equilibrium of (Dove, Dove), not (Hawk, Dove) or (Dove, Hawk) as the research of Blazquez de Paz et al. suggested.

8 Discussion

8.1 Unexpected intermediate results

Our hypotheses and our research were based on the assumption of the formation of a Nash equilibrium. However, our results did not show such equilibria. Thus, the ground of our whole research conflicts with our results. Nonetheless, we believe that our answer to the research question is still valid, as it is not depending on the idea of a Nash equilibrium. It does mean, however, that our model and thus our simulations might be incorrect in some way.

8.2 Applying learned values

Following this conclusion, we hope we can apply some of the lessons learned in this research to real-life scenarios as well. Seeing how your initial stance influences the rest of the proceeding conflict, could lead you to change that stance, to try and obtain a more optimal result for yourself.

As mentioned in section 2.1, Hawk-Dove has been used to analyse past conflicts [LHQA22]. Our study, which analyses resulting equilibria, could make it possible to model and simulate future conflicts. Therefore, we hope it could also help with conflict avoidance. If beforehand you can already make an accurate simulation of how a conflict will eventually resolve itself, there is no need to enter the actual conflict in real life. We think there are enough conflicts in this world as it is, you only have to look at what is happening on the east side of our own continent, so we think any means of reducing them should be encouraged as much as possible.

8.3 Follow-Up research

It could also be very interesting to expand upon this research. Because we only had limited time and resources, we have had to keep our scope quite limited for now, which makes one wonder what would happen if that is not the case. The first thing that would be interesting to follow up on, is to increase the number of iterations of both the inner and the outer loop. This would require a lot of computing power and time but could lead to valuable new insights. Increasing the number of iterations of the inner loop could help identify more outliers, which could influence the mean calculated in the outer loop. Increasing the number of iterations of the outer loop could provide more accurate results, because the randomness of the outcomes will have less of an impact on the mean, meaning the data more accurately represents the actual behaviour of the Agents.

Secondly, future research should verify the model to see why no Nash Equilibrium occurs. If our model is suggested or proven to be incorrect, our conclusion and results might thus be affected. This is of big importance.

Another thing to experiment with is the ratio of the conflict costs compared to the reward. It could be interesting to see if being too aggressive would be punished way less or more than it is now, which could have significant influences on the results. In a similar fashion, we could also look at other variants of the game, as discussed in the introduction in section 1.2. This would entail changing the current payoff matrix.

A final proposal we have for future research is that the Agents could declare their intention to the other before choosing their final stance. However, the Agents wouldn't necessarily have to follow this intention. The behaviour of the Agents would therefore have to change, to take these intentions into account before choosing a final stance. We think this could be interesting, because it more closely follows real-life conflicts, meaning the gathered data could prove to be more useful when using it for conflict resolution.

References

[AdS98] P. Auger, R.Bravo de la Parra, and E. Sánchez. Hawk-dove game and competition dynamics. *Mathematical and Computer Modelling*, 27(4):89–98, 1998.

[dPK22] M. Blazquez de Paz and N. Koptyug. Equilibrium selection in hawk-dove games. *SSRN*, 2022.

[JBZ20] Zhigang Ji, James Brown, and Jianfu Zhang. True random number generator (trng) for secure communications in the era of iot. In *2020 China Semiconductor Technology International Conference (CSTIC)*, pages 1–5, 2020.

[KGP14] Hanna Kokko, Simon C. Griffith, and Sarah R. Pryke. The hawk–dove game in a sexually reproducing species explains a colourful polymorphism of an endangered bird. *The Royal Society*, 2014.

[LHQA22] Wenlong Li, Shupei Huang, Yabin Qi, and Haizhong An. Rdeu hawk-dove game analysis of the china-australia iron ore trade conflict. *Resources Policy*, 77:102643, 2022.

[MP73] J. Maynard Smith and G. R. Price. The logic of animal conflict. *Nature*, 246(5427):15–18, 1973.

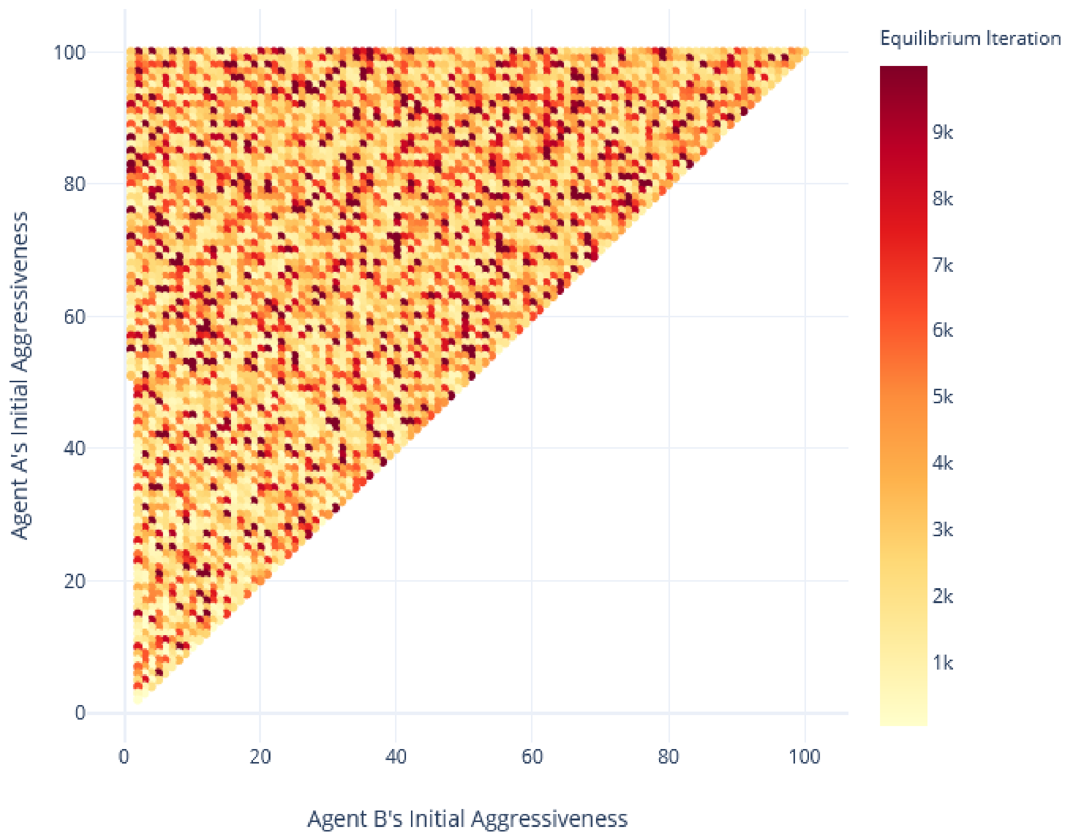
[Plo] Scatter chart maker - plotly chart studio. *https://chart-studio.plotly.com/create/scatter-chart//*.

[PRB23] Silas Peters, Fabian Rutten, and Olaf Boekholt. Hawk-dove simulation software. *https://github.com/SilasPeters/Hawk-Dove*, Apr 2023.

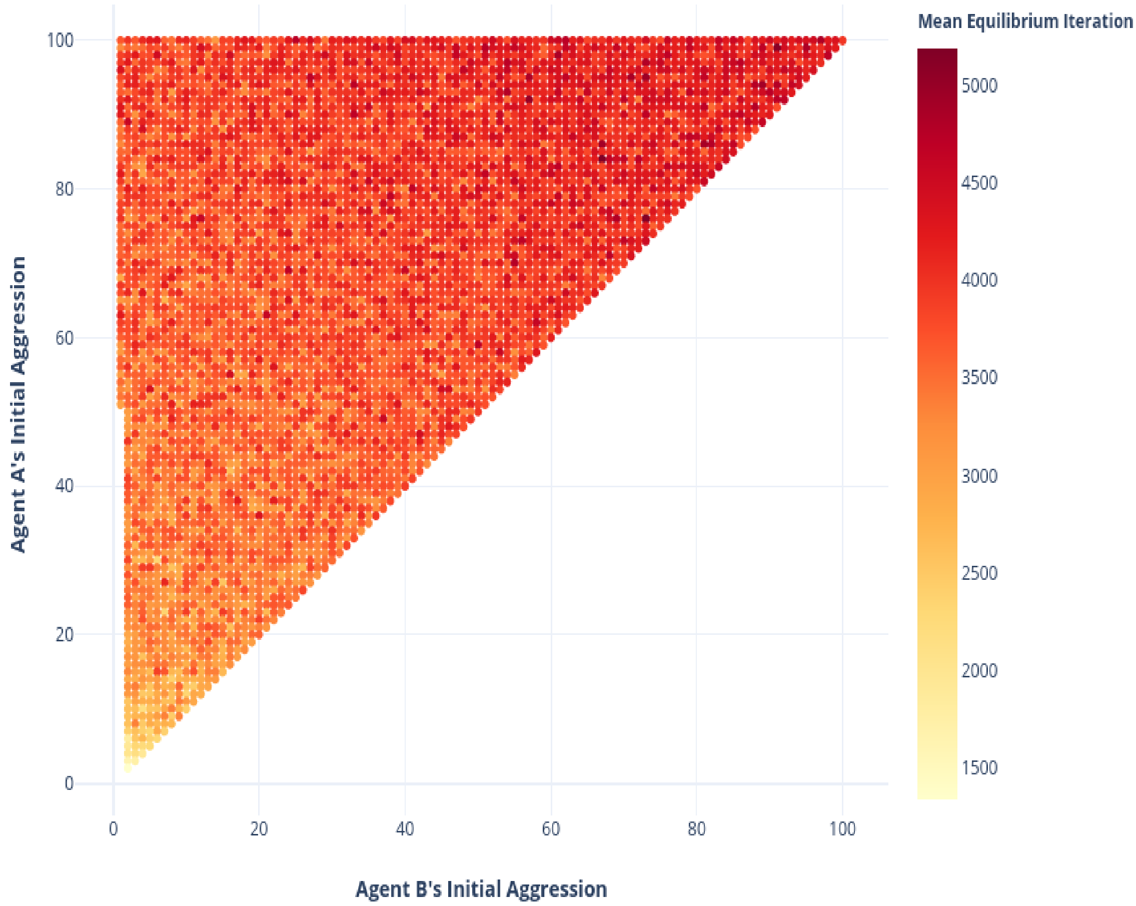
[WIL60] J.H. WILKINSON. Error analysis of floating-point computation. *Numerische Mathematik*, 2:319–340, 1960.

A Graph with One Random Test

$Seed : 283900703$



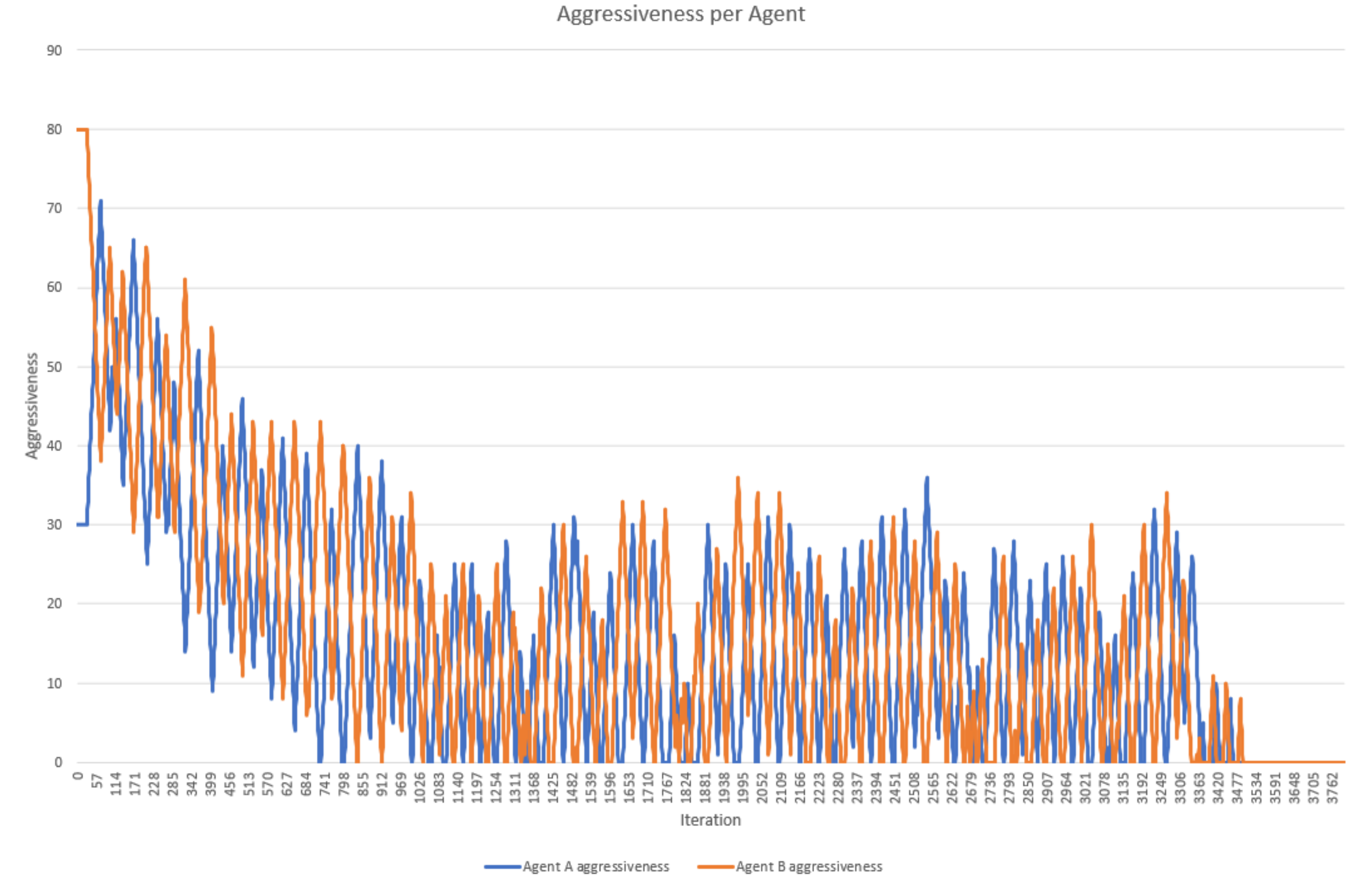
B Results Graph



C Aggressiveness Graph

Seed : 4694968

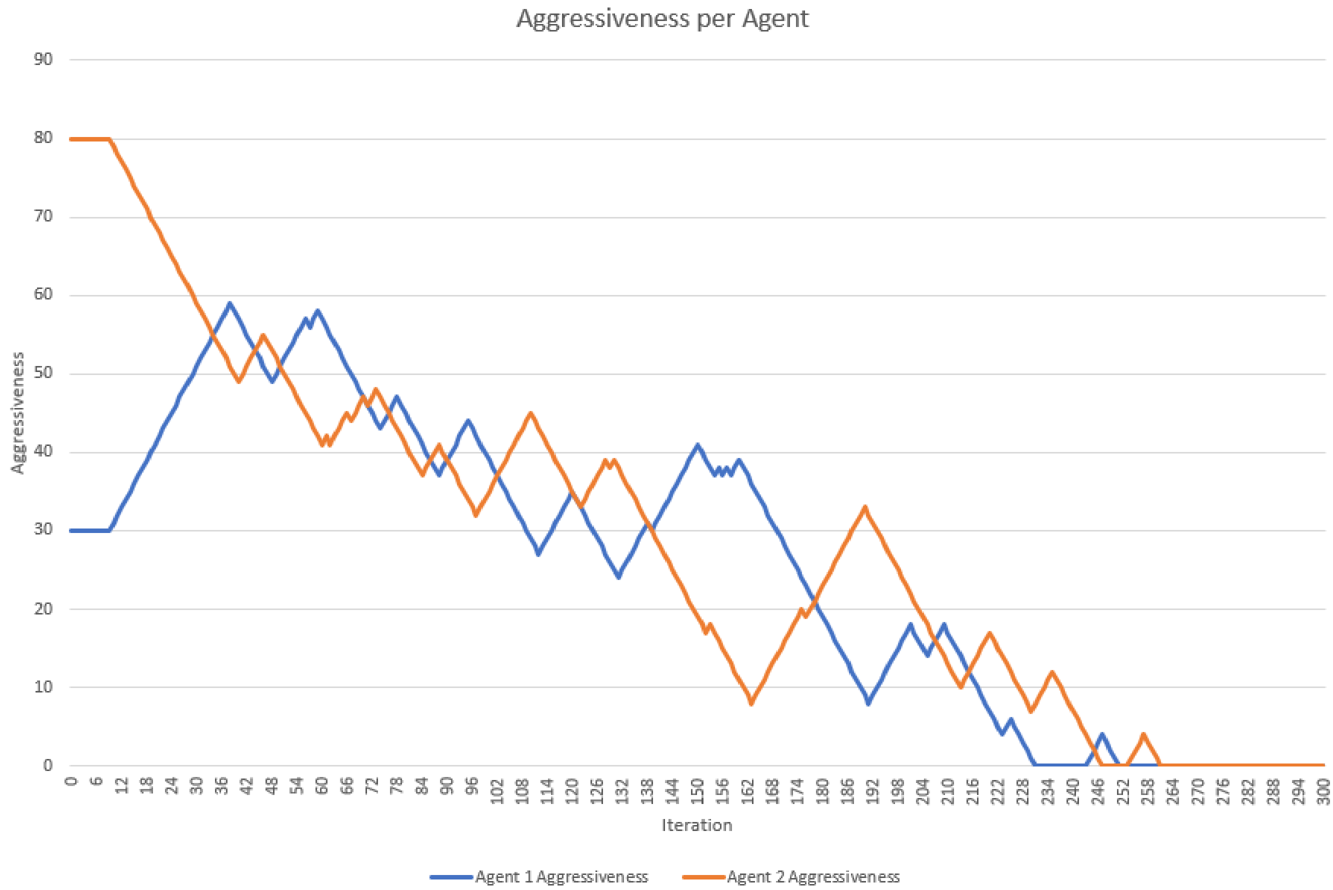
Agent A's Initial Aggressiveness: 30
Agent B's Initial Aggressiveness: 80
History Range per Agent: 30



D Aggressiveness Graph

Seed : 4694968

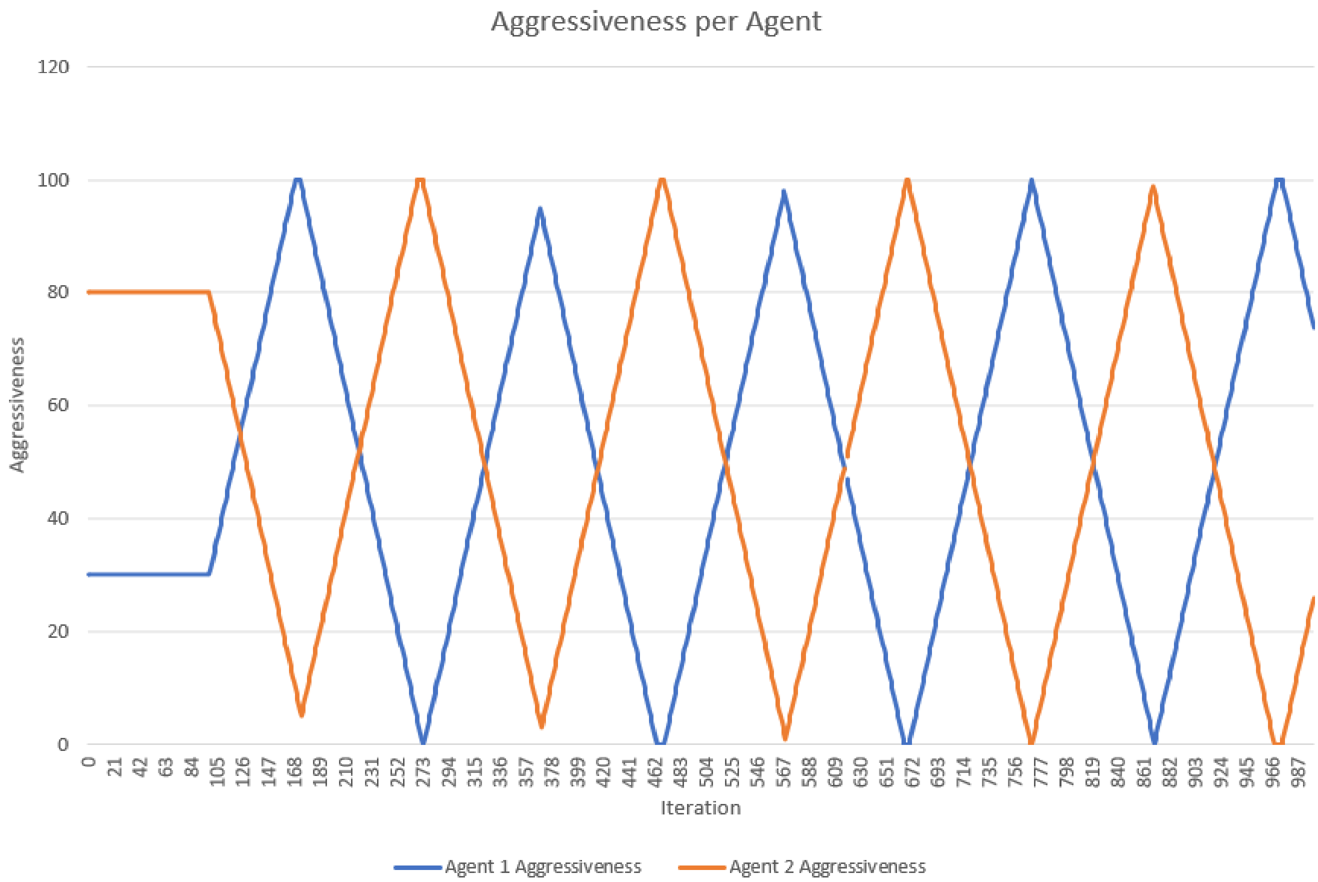
Agent A's Initial Aggressiveness: 30
Agent B's Initial Aggressiveness: 80
History Range per Agent: 10



E Aggressiveness Graph

Seed : 4694968

Agent A's Initial Aggressiveness: 30
Agent B's Initial Aggressiveness: 80
History Range per Agent: 100



F Values not represented in Appendix B

Format within every column:

`Initial Aggressiveness Agent A - Initial Aggressiveness Agent B - Mean Equilibrium Iteration`

1 - 50 - 3849.24
1 - 49 - 3681.97
1 - 48 - 3644.75
1 - 47 - 3298.86
1 - 46 - 3342.88
1 - 45 - 3324.48
1 - 44 - 2842.21
1 - 43 - 3751.55
1 - 42 - 3118.69
1 - 41 - 2974.45
1 - 40 - 3340.34
1 - 39 - 3468.16
1 - 38 - 3392.36
1 - 37 - 3388.58
1 - 36 - 3191.95
1 - 35 - 3483.46
1 - 34 - 3251.19
1 - 33 - 3403.59
1 - 32 - 2858.93
1 - 31 - 3561.89
1 - 30 - 3238.35
1 - 29 - 3402.08
1 - 28 - 3822.7
1 - 27 - 2849.7
1 - 26 - 3043.12
1 - 25 - 3147.55
1 - 24 - 3321.28
1 - 23 - 3472.65
1 - 22 - 3404.68
1 - 21 - 3508.29
1 - 20 - 2708.69
1 - 19 - 3201.6
1 - 18 - 3333.53
1 - 17 - 3334.19
1 - 16 - 3536.33
1 - 15 - 3286.16
1 - 14 - 2984.86
1 - 13 - 2736.05
1 - 12 - 3005.32
1 - 11 - 2884.56
1 - 10 - 3031.45
1 - 9 - 2827.24
1 - 8 - 2559.77
1 - 7 - 2652.51
1 - 6 - 2712.46
1 - 5 - 2278.32
1 - 4 - 2220.21
1 - 3 - 1697.97
1 - 2 - 1409.78
1 - 1 - 872.66
0 - 100 - 3694.81
0 - 99 - 4023.16
0 - 98 - 3330.15
0 - 97 - 3970.9
0 - 96 - 4121.1
0 - 95 - 3366.58
0 - 94 - 3798.46
0 - 93 - 3921.83
0 - 92 - 3583.11
0 - 91 - 3972.47
0 - 90 - 3615.78
0 - 89 - 3676.27
0 - 88 - 3315.49
0 - 87 - 3957.88
0 - 86 - 3812.08
0 - 85 - 4033.99
0 - 84 - 3757.87
0 - 83 - 4252.84
0 - 82 - 3988.98
0 - 81 - 3773.53
0 - 80 - 3247.12
0 - 79 - 3972.11
0 - 78 - 3187.7
0 - 77 - 3702.78
0 - 76 - 3494.61
0 - 75 - 3916.2
0 - 74 - 3293.77
0 - 73 - 3277.79
0 - 72 - 3623.09
0 - 71 - 3473.58
0 - 70 - 3643.46
0 - 69 - 3433.3
0 - 68 - 3694.37
0 - 67 - 3350.13
0 - 66 - 4179.43
0 - 65 - 3730.06
0 - 64 - 3919.13
0 - 63 - 3204.07
0 - 62 - 4322.26
0 - 61 - 3794.47
0 - 60 - 2961.81
0 - 59 - 3399.06
0 - 58 - 3777.49
0 - 57 - 3170.04
0 - 56 - 3329.53
0 - 55 - 3559.26
0 - 54 - 3480.81
0 - 53 - 3753.56
0 - 52 - 3416.7
0 - 51 - 3741.23
0 - 50 - 3501.8
0 - 49 - 3066.41
0 - 48 - 3008.64
0 - 47 - 3015.57
0 - 46 - 2898.57
0 - 45 - 3315.71
0 - 44 - 3584.56
0 - 43 - 3760.23
0 - 42 - 3342.1
0 - 41 - 3509.66
0 - 40 - 3068.89
0 - 39 - 2903.93
0 - 38 - 3142.75
0 - 37 - 3535.78
0 - 36 - 3334.27
0 - 35 - 3382.77
0 - 34 - 3252.71
0 - 33 - 3447.3
0 - 32 - 3192.64
0 - 31 - 3688.67
0 - 30 - 3367.19
0 - 29 - 3330.09
0 - 28 - 3737.94
0 - 27 - 3497.04
0 - 26 - 2821.32
0 - 25 - 2521.29
0 - 24 - 3545.43
0 - 23 - 3495.53
0 - 22 - 3052.73
0 - 21 - 3055.85
0 - 20 - 2829.35
0 - 19 - 3012.26
0 - 18 - 3487.79
0 - 17 - 2732.33
0 - 16 - 3059.59
0 - 15 - 2787.13
0 - 14 - 3207.58
0 - 13 - 3168.81
0 - 12 - 2709.99
0 - 11 - 2359.12
0 - 10 - 3059.95
0 - 9 - 2822.99
0 - 8 - 2377.77
0 - 7 - 2427.29
0 - 6 - 2366.98
0 - 5 - 1572.84
0 - 4 - 2004.44
0 - 3 - 1302.11
0 - 2 - 961.34
0 - 1 - 425.73
0 - 0 - 0