Game Theory: An Application in Behavioral Simulation

Arman Luthra , Walid Kasab , Abdul Basit Tonmoy

Wabash College

Abstract

Our project explores interactions of game theory-based strategies among virtual creatures competing for food resources through simulations. Three primary strategies are examined: cooperative Doves that share resources equally, competitive Hawks that aggressively monopolize food, and newly introduced social Vultures that thrive in groups but perish alone. The simulation varies encounters between these strategies, observing survival and reproduction outcomes under different conditions. Modifications to the payoff matrix alter the different creatures' survival probabilities when conflicting with one another. The results demonstrate how cooperative versus competitive behaviors impact populations, showing an application of game theory and stochastic simulation through an ecological simulation model.

KEYWORDS: GAME THEORY, BEHAVIORAL SIMULATION, POPULATION EQUILIBRIUM, VIRTUAL ECOLOGICAL ECOSYSTEM, HAWKS & DOVES

1 Introduction

For our project, we look at how virtual creatures behave when they are competing for food resources [1][2]. We use some ideas from game theory[3] to set up different strategies these virtual creatures can follow. In our simulation, food appears in pairs each day, and the creatures randomly move towards one of those food pair locations. When multiple creatures end up at the same food source, we see how their pre-programmed strategies determine how they split up or fight over those food units.

We test two main strategies: "Doves" who share the food equally, and "Hawks" who try to bully others and take as much food as possible for themselves. The rules are:

- Eating 1 food unit lets a creature survive to the next day
- Eating 2 food units lets a creature survive and reproduce

For Doves, if they meet at a food pair, they simply split it 50/50 so they each get 1 unit and survive. Hawks get aggressive when they meet a Dove. The Hawk goes for both food units, while only letting the Dove have scraps $\left(\frac{1}{2}\text{ a unit}\right)$. So the Hawk survives and maybe reproduces, while the Dove only has a 50% chance of surviving. When two Hawks meet, they expend a ton of energy fighting over the food pair. They each get 1 unit, but use up all that energy from the intense fight, so neither survives that day.

In addition to this base model, we introduce a new creature, which we will call "Vultures". We pre-define the character of the vultures as follows:

- When a vulture meets another vulture at a food pair, they reproduce.
- The vultures are extremely social creatures. They can't live alone. Because of that, when a vulture is alone at a food pair alone, it dies.
- When a vulture meets a dove at a food pair, the vulture reproduces and the dove survives.

- The vultures hate the hawks. So, when a vulture meets a hawk, they fight. As vultures are more powerful than hawks, in every case, the vultures kill the hawks.

By running this simulation with creatures following the Hawk, Dove and Vulture strategies, we observe what happens when individualistic and cooperative strategies co-exist and compete over resources. We also change the payoff matrix to see how that affects the equilibrium.

As for the division of work on the project, Walid and Arman worked on the baseline model. For the changing payoff matrix and the code regarding Vultures, Walid and Abdul wrote that. The Presentation was made by Arman and Abdul, and the report was written by all three, and then compiled in Latex by Arman.

2 Model

2.1 Baseline Case[4]

Theoretically, equilibrium is achieved when the addition of doves and hawks is equal, such that the ratio of the doves to the whole population becomes almost constant over time.

$$\textit{Addition of Doves} = \textit{Addition of Hawks}$$

We can get the equilibrium point by solving the following equation based in the payoff matrix given in Figure 2.1.

	Dove	**Hawk**
Dove	1,1	$\frac{1}{2}, \frac{3}{2}$
Hawk	$\frac{3}{2}, \frac{1}{2}$	0,0

Figure 2.1: Baseline Payoff Matrix

For simplification, let us call the Addition of Doves and Hawks D and H respectively. Also, let us call the ratio of doves and hawks d and h respectively.

$$D = H$$

$$d \cdot 1 + \frac{1}{2} \cdot h = d \cdot \frac{3}{2} + 0 \cdot h$$

Since our equation has 2 variables, we can use the substitution method. This is possible since we know the relationship between d & h.

$$\because\ 1 = d + h \therefore\ h = 1 - d$$

$$d \cdot 1 + \frac{1}{2} \cdot (1 - d) = d \cdot \frac{3}{2} + 0 \cdot (1 - d)$$

$$d = 0.5 \quad , \quad h = 0.5$$

2.2 Change in Payoff Matrix

For the 2nd scenario, we change the probability of survival of hawks when two hawks meet, changing it from 0% to 25%, based on the payoff matrix given in Figure 2.2.

	Dove	**Hawk**
Dove	1,1	$\frac{1}{2}, \frac{3}{2}$
Hawk	$\frac{3}{2}, \frac{1}{2}$	$\frac{1}{4}, \frac{1}{4}$

Figure 2.2: Altered Payoff Matrix

Making the corresponding changes to the d and h values, we obtain the following expected values for the equilibrium:

$$d \cdot 1 + \frac{1}{2} \cdot (1-d) = d \cdot \frac{3}{2} + \frac{1}{4} \cdot (1-d)$$
$$d = 0.\overline{3} \quad , \quad h = 0.\overline{6}$$

2.3 Prisoner's Dilemma Case

For the 3rd scenario where we change the probability of survive of hawks when two hawks meet from 0% to 75% , this turns the whole situation to prisoner's dilemma

$$d \cdot 1 + \frac{1}{2} \cdot (1-d) = d \cdot \frac{3}{2} + \frac{3}{4} \cdot (1-d)$$
$$d = -1 \quad , \quad h = 2$$

This is what the payoff matrix looks like for this situation:

	Dove	**Hawk**
Dove	1,1	$\frac{1}{2}, \frac{3}{2}$
Hawk	$\frac{3}{2}, \frac{1}{2}$	$\frac{3}{4}, \frac{3}{4}$

Figure 2.3: Prisoner's Dilemma Case

Since one of the values is negative, there isn't any equilibrium. This observation is further validated since on plotting the graphs of the two sides of the equation, we observe that the values never intersect in the 1st quadrant. They still attempt to reach the equilibrium point even though the value for the ratio of doves is a negative which results in the doves going extinct.

3 Methods

3.1 Baseline Case

The core logic of our simulation is contained in the `simulatePopulations` function. After initializing global parameters and helper functions, it enters the main loop that iterates over the specified number of days, as long as there are still living hawks or doves.

Function Attributes

```
simulatePopulations[initialDoves_, initialHawks_, numLists_,
    numIterations_, printing_: 0]
```

Each day, it first initializes variables that track the counts of different hawk/dove interaction outcomes for the next day. It then loops through randomly selecting available hawks (2) and doves (1) from the remaining population counts using `RandomChoice`, and it is weighted. For each selected creature, it calls `addToList` which uses another internal loop to find a food pair location (represented as a list) that is not already full with 2 creatures. It randomly picks one of these available locations and appends the hawk/dove identifier to that list.

Once all creatures are placed at the food pair locations for the day, `simulatePopulations` calls `processInteractions`. This uses a nested Do loop to iterate through each food pair location list. It employs a Switch statement to analyze the composition of hawks and doves at that location and apply the prescribed hawk/dove interaction rules. For example, if the list contains 1, 2, representing a dove and a hawk, it uses `RandomReal` to determine if the hawk gets 1 or 2 food units based on the 50% chance in the rules. The dove's food intake is also randomly assigned 0 or 1 accordingly. Counters like `oneDove`, `twoHawks` etc. track the outcomes.

`processInteractions` Structure

```
processInteractions[pairs_] := Module[{newDoves = 0, newHawks = 0, u=
    RandomReal[]},
        Do[
            Switch[pair,
                {1}, newDoves += 2; oneDove++,
                {2}, newHawks += 2; oneHawk++,
                {1, 1}, newDoves += 2; twoDoves++,
                {2, 2},Null; twoHawks++,
                {1, 2} | {2, 1},
                If[RandomReal[] < 0.5, newHawks += 1, newHawks += 2];
                If[RandomReal[] < 0.5, newDoves += 0, newDoves += 1];
                doveAndHawk++
            ],
            {pair, pairs}
        ];
        {newDoves, newHawks}
    ];
```

The `processInteractions` function returns the total new dove and hawk counts after resolving all interactions. These are stored (for the next iteration) and used to `appendToTable` which logs simulation data for that day to the `printTable`. Before moving to the next day, `initializeLists` resets all food pair locations to empty lists. After the specified number of days, the simulation ends. If configured to print, it outputs the `printTable` showing daily dove, hawk, and hawk/dove ratio counts, as well as the split of interaction outcomes that drove the population dynamics.

For figuring out what the resulting equilibrium was, we used the following code and compared the results with the theoretical equilibrium to check the validity of our approach.

Extracting Final Ratios

```
data := Last[simulatePopulations[10, 10, 100, 150, 0][[1]][[2;;-1, 4]]]
```

3.2 Change in Payoff Matrix

We change the payoff matrix as shown in Figure 2.2 so that if two hawks meet their probability of surviving is 25%. We edit the code to meet the new payoff matrix by editing line 7 in the `processInteractions` function to be like the following:

Changes in the Switch statement

```
....
{2, 2}, If[RandomReal[] > 0.75, newHawks += 1];
If[RandomReal[] > 0.75, newHawks += 1];twoHawks++
....
```

So, the probability of a hawk surviving will happen if `RandomReal[]` is more than 0.75, which occurs 25% of the time.

3.3 Prisoner's Dilemma Case

We change the payoff matrix as shown in Figure 2.3 so that if two hawks meet their probability of surviving is 75%. Similar to the procedure followed in scenario 2, we edit line 7 in processInteractions to be :

Changes in the Switch statement

```
....
{2, 2}, If[RandomReal[] > 0.25, newHawks += 1];
If[RandomReal[] > 0.25, newHawks += 1];twoHawks++
....
```

So, the probability of a hawk surviving will happen if `RandomReal[]` is more than 0.25, which happens 75% of the time.

3.4 Simulating a New Species

Basing this off out baseline model, we initialize a new variable: the vultures as `Vs`, and the `Ratio of V` , which is defined as,

$$\frac{\text{population of Vultures}}{\text{Total Doves} + \text{Total Hawks} + \text{Total Vultures}}$$

Then we change the `processInteraction` module where we add new conditions which reflects the characteristics of the vultures we explained earlier.

processInteraction

```
{3}, Null; oneV++,
.....
{3, 3}, newV += 3; twoV++,
{3, 2} | {2, 3}, newV += 1; VH++,
{3, 1} | {1, 3}, newV += 2; newDoves++; VD++,
....
```

After that, outside the module and inside the for loop we added another condition where we check if `V`>0. Then we simulate the code by changing different inputs.

4 Results

We setup the parameters as `simulatePopulations[10,10,100,250,0]`, which translates to 10 doves and hawks each, 100 pairs of food and 250 iterations, and printing set to off. Upon running the simulation and plotting the number of hawks and doves, we achieve the following results for the different cases.

4.1 Baseline Case

Observing Figure 4.1, the ratio of doves and hawks comes close to the line $y = 0.5$ at the end of the simulation.

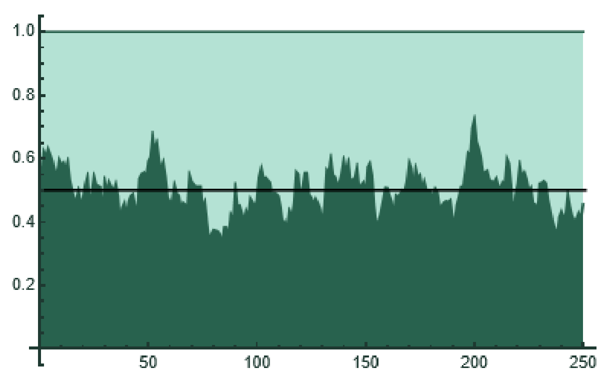

Figure 4.1: Base Case

After running this simulation 100 times, and then finding the mean of the ratio of hawks and doves at the end of the simulation, we find the mean to be 0.487 ± 0.014 (95% CI). The confidence interval houses the value 0.5, which validates with our simulation since it matches with our theoretical results achieved in Section 2.1.

4.2 Change in Payoff Matrix

Observing Figure 4.2, the ratio of doves and hawks comes close to the line $y = \frac{1}{3}$ at the end of the simulation. The simulation results in a mean of 0.319 ± 0.015 (95% CI). for the ratio between doves and hawks at the end of the simulation, which further endorses the theoretical value of 0.3 we found in Section 2.2.

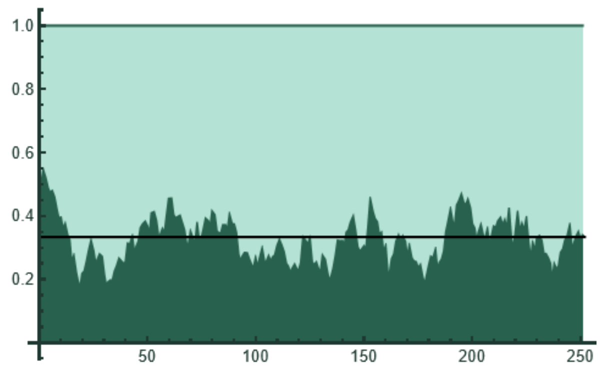

Figure 4.2: Simulation with updated Matrix

4.3 Prisoner's Dilemma Case

The simulation results (Figure 4.3) match with the theoretical value of 0 we found in Section 2.3 for the ratio of doves and hawks, since the doves go extinct. This also corresponds with the Prisoner's Dilemma game theory as we observe the survival of the ones with the fittest strategy.

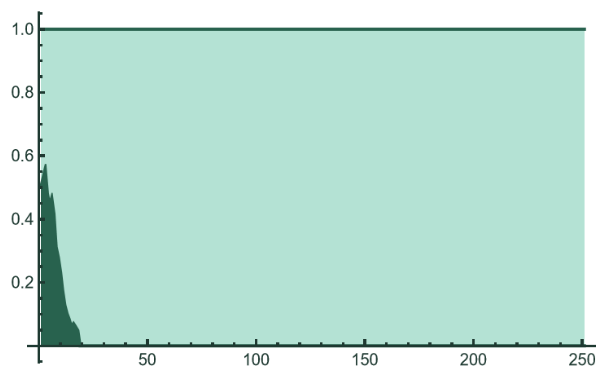

Figure 4.3: Prisoner's Dilemma Case

4.4 Simulating a New Species

The graph in Figure 4.4 implies that when there are more food pairs available, vultures go extinct. This phenomena occurs because of how we coded vultures' to have extremely social nature. With an increasing number of food pairs, there is a greater likelihood of just one vulture at each food pair, resulting in their extinction.

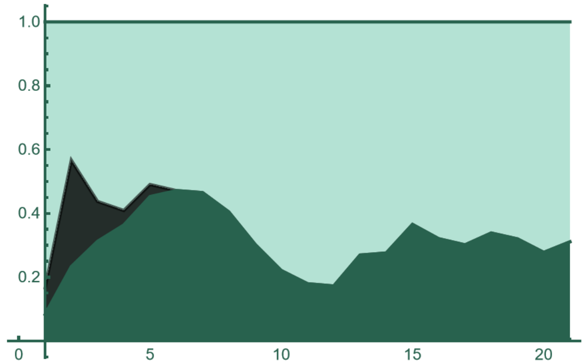

Figure 4.4: Scenario with more Food Pairs

In contrast, when there are fewer food pairs, vultures breed quickly. Hawks and doves eventually become extinct as a consequence of the lack of food. This happens because having fewer food pairs increases the possibility of two vultures occupying each one, which speeds up their reproduction. As a result, the growing vulture population restricts doves' and hawks' access to food pairs, eventually leading to food shortages and extinction.

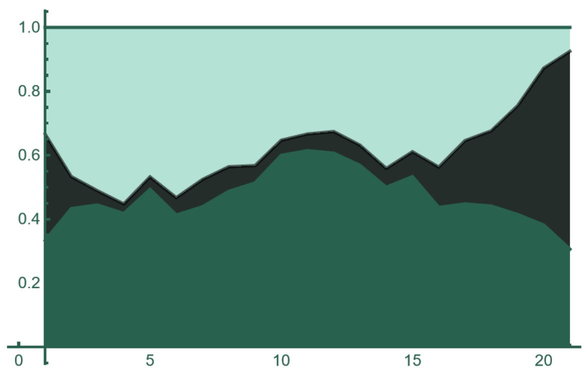

Figure 4.5: Scenario with less Food Pairs

5 Conclusion

We are able to successfully simulate the baseline hawk and dove strategies, matching the expected theoretical equilibrium points in our simulation. Scenarios 2.2 and 2.3 also align with expectations. Additionally, we edit the code to introduce a new "Vultures" creature with unique behavioral rules when interacting with hawks and doves over contested food resources. Upon reviewing our approach, we realize our simulation effectively implements a Markov chain model. Each day's iteration depends solely on the previous day's hawk, dove and vulture population counts. The `processInteractions` function acts as the transition model, determining the next day's counts probabilistically based on the current composition.

6 References

[1] K. Prestwich. ““hawk vs. dove: An introduction to simulations of ess".” Accessed: 01 05 2024. (2009), [Online]. Available: https://college.holycross.edu/faculty/kprestwi/behavior/ESS/HvD_Sim_Intro.html.

[2] L. R. Izquierdo. ““an imitation model in the hawk-dove game".” Accessed: 01 05 2024. (2010), [Online]. Available: https://demonstrations.wolfram.com/AnImitationModelInTheHawkDoveGame/.

[3] W. contributors. ““chicken (game)".” Accessed: 01 05 2024. (2023), [Online]. Available: https://en.wikipedia.org/wiki/Chicken_(game).

[4] Primer. “"simulating the evolution of aggression".” Accessed: 01 05 2024. (2019), [Online]. Available: https://www.youtube.com/watch?v=YNMkADpvO4w.