Game theoretic analysis to show parasitic cooperation at traffic intersections that encompass conflict

A thesis submitted to Indian Institute of Science Education and Research Pune in partial fulfillment of the requirements for the BS-MS Dual Degree Programme

Thesis Supervisor: Dr. Pranay Goel

by

Ankita Sharma

April 2, 2014

Indian Institute of Science Education and Research
Sai Trinity Building, Pashan, Pune India 411021

Certificate

This is to certify that this thesis entitled "Game theoretic analysis to show parasitic cooperation at traffic intersections that encompass conflict" submitted towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research Pune represents original research carried out by **Ankita Sharma** at **Indian Institute of Science Education and Research, Pune**, under the supervision of **Dr. Pranay Goel**, Dept. of Mathematics and Biology, IISER Pune during the academic year 2013-2014.

Thesis Advisory Committee
Dr. Pranay Goel
Dr. M.S.Santhanam

Dr.A.Raghuram
Coordinator of
Maths Faculty

Abstract

This study investigates the existence of cooperation at road intersections where there is no enforcement. We recorded various videos of a road intersection near Mahabaleshwar hotel on Baner-Aundh road Pune, as evidence to the fact that people tend towards cooperation even though there were no traffic lights. We have designed various games and have written simulations to calculate the payoffs of the cars involved in the games. On adding an extra car to the standard snow drift game, the payoffs indicate the emergence of parasitic cooperation on a road intersection. The payoffs from these games have been compared with the payoffs calculated in a standard snow drift game and it concludes that cars have a better chance of crossing the intersection if they cooperate.

Acknowledgements

It is with immense gratitude that I acknowledge the support and help of my guide Dr. Pranay Goel. I am thankful for his composed nature, his motivation, encouragement, enthusiasm, his useful comments, remarks and engagement through the learning process of this master thesis.

I would like to take this opportunity to thank IISER Pune for providing funding for the project work without which the project would not have been possible.

I would also like to thank all my friends at IISER Pune, Varun Karamshetty, Aditi Jakhar, Dhanashree Khanale, Deepika Ananda and Purvi Tiwari for their support all year long.

Contents

Chapter 1

Introduction

When it comes to explaining evolution of human cooperation, the observed degree of cooperation is generally higher than predicted. We can see examples of selfless help in large number around us. Effect of cooperation especially on humans has always been very intense. If we talk about the theory of survival of the fittest, we know that evolution is based a lot on competition among individuals. Every individual should be designed in a way to increase its own payoff even if it is at the cost of the rival. Since payoff in evolution is measured in terms of fitness, average fitness appears to be more of a defector than that of a person who cooperates. If an individual is really selfish and pays the cost only for himself without compromising its own fitness for others then average fitness of a defector seems to be more than that of a cooperator. If there is a community of both defectors and cooperators, and since average fitness of defectors keeps going higher with time then natural selection should lead to a large community of defectors.

We know from Prisoner's dilemma, that in order to win in a competition, the better choice is to defect and gain the maximum payoff because we do not know what the other player might do. In the second year at IISER Pune, in one of our behavior biology labs we played a game about Tragedy of commons with chocolates. About 20 chocolates were kept on a table and a group of people were asked to pick as many chocolates as they like. After each game, the number of chocolates left behind on the table were doubled. Every time the game was played, there were some defectors who would grab on to as many chocolates as possible and there were very few cooperators who could foresee that this game could go on to infinite chocolates. Since we were not allowed to talk to each other before the game, there was always a fear of other player defecting in the game which would lead us to even less number of chocolates. None of the games lasted for more than three times because someone or the other switched from cooperation to defection in order to increase payoff.

Even though defection seems to result in higher payoff for an individual, people tend to cooperate. If we consider a crowded road intersection with no traffic light or traffic policeman, everybody seems to cooperate in crossing the road. But if every single person is expected to be selfish, such intersections should always result into a traffic jam,. But we know this situation of traffic jams is not that prevalent. Rather it can be observed that even on a busy road with heavy traffic moving from all sides, people tend to stop and allow others to get across.

This high degree of cooperation at traffic intersections gives the key problem of this project. If people are bound to be selfish and these situations are always suppose to result in traffic jams, then how does it not occur at every such intersection? Why is there such high level of cooperation? And if it is not cooperation that occurs then how do people get across on a road intersection with no traffic light or traffic policemen? The aim of this project is to explain such behaviors by designing and analyzing various games that would explain this cooperation among people at such road intersections.

The idea of this problem comes from a road intersection on Baner-Aundh road in Pune and is close to IISER campus. This is the intersection next to Symantec Building on Baner Road. This is one of those intersections that faces heavy traffic each day. Every day, during office hours, this road is heavily packed with vehicles and still each one of the vehicles, with minimal cooperation, is able to get across without any traffic signal or a traffic policeman to guide them.

There is another similar intersection quite near to this site near Mahabaleshwar hotel on the Baner road itself which has also helped us develop various games for the project. We have recorded videos at these two traffic intersections which work as evidence for different theories that come up while analyzing the problem. The following image, i.e. Figure 1.1, shows the intersection near Mahabaleshwar hotel. The image shows a bunch of people waiting in a cross lane trying to cross through the heavy traffic from the main lane. The next picture, i.e. Figure 1.2, is a Google map to this intersection near Mahabaleshwar hotel. The red balloon in the image is where hotel Mahabaleshwar is located. While the two large pink arrows drawn on the picture show the intersection. The two arrows point in the direction of the movement of traffic at this intersection.

Figure 1.1: The intersection near Mahabaleshwar hotel on Baner-Aundh road, Pune. The image shows a bunch of vehicles gathered at the intersection to cross the road together.

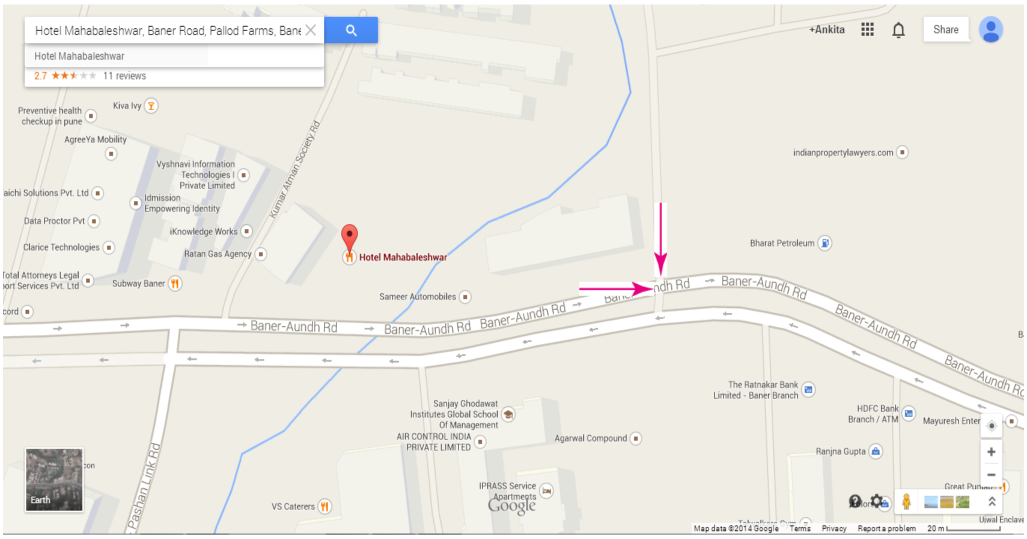


Figure 1.2: Google map for the intersection near Mahabaleshwar hotel. The red balloon shows the location of Mahabaleshwar hotel on Baner-Aundh road in Pune. The two pink arrows show the direction of traffic from the two lanes of the intersection. The arrow pointing upwards is the main lane where there is continuous flow of traffic and the other arrow shows the direction of the cross lane where there are vehicles coming out to cross the main lane.

Chapter 2

Game theoretic analysis of a road intersection

A standard snow drift game is game played between two players. There is a pile of snow and two players stand on either side of the snow. They both need to get their respective cars across. The only way to go ahead is to clear the pile of snow in between. One of the possible solutions is that the first player clears the snow, in which case the first player loses payoff increasing the payoff of the second player. Another possibility is if the second player cooperates and clears the snow while the first one sits. Third possibility would be if they both cooperate making the payoff of both lesser but equal. The last case would be if they both sit and lose payoff as no one cleared the snow and hence no one can move forward. The payoff of an individual will be maximum in the case when one defects while the other cooperates.

2.1 Representing a road intersection in the form of a matrix

In order to analyze a road intersection, we have designed games where cars trying to cross the intersection are our players and the intersection is where the game is being played. We represent our games by way of matrices. This matrix form representation of a game includes all possible strategies for the players involved. Figure 2.1 shows the intersection on Baner-Aundh road near Mahabaleshwar hotel in Pune. A matrix has been drawn on top of the road to show how we have made a game out of a road intersection. The matrix shows various positions where a car can exist by way of arrows. The top most position of the matrix has been marked as (1,1) so that the other positions in the matrix can be represented accordingly.

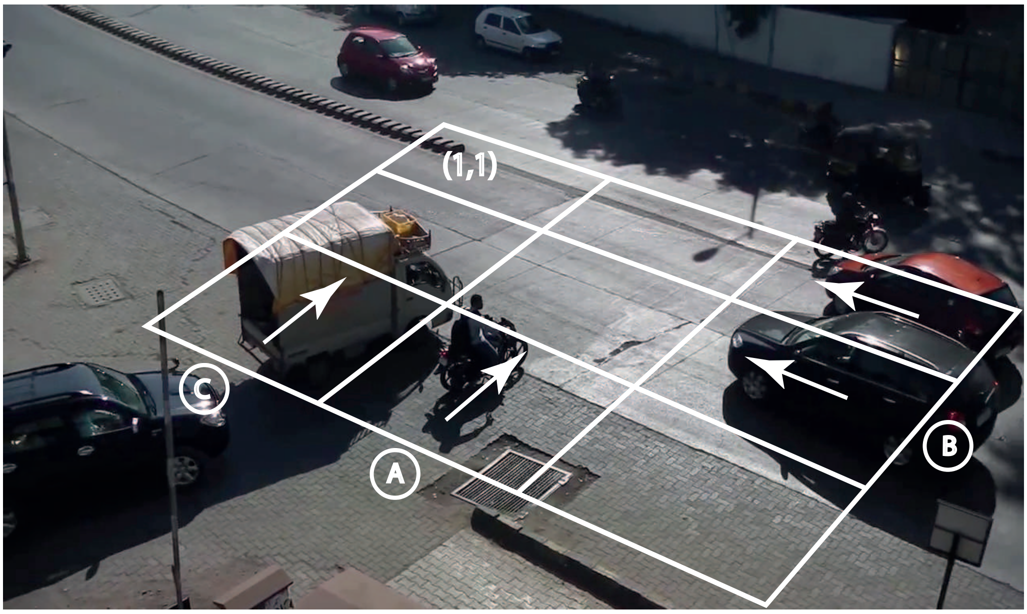


Figure 2.1: A matrix has been projected on top of the intersection near Mahabaleshwar hotel at Baner-Aundh road, Pune. The arrows represent cars and they point in the direction of the movement of that car. The lane from which vehicles marked as A and C originate is the 'cross lane' while the other road which has a black car marked as B is the 'main lane'. The top most square in the matrix has been marked as (1,1).

The two vehicles marked as A and C in the image are emerging from a cross lane of this intersection and the lane they are trying to cross is the main Baner-Aundh road. The games have been designed based on this structure of the intersection. If the top most position of the matrix is (1,1) then the center position becomes (2,2). It becomes clear from the image that vehicles A and B are competing for position (2,2). While vehicle marked as C can easily move to next position, that is (1,2) in order to cross the main lane. This way, we are able to design a game of three cars placed at positions (3,2), (2,3) and (3,1) in a matrix trying to cross through the matrix. The cars are being called A, B and C, just like the vehicles in the image. It is assumed that vehicles on this intersection are moving one step at a time of this matrix in order to cross the road. If there is a car at position (3,1) that is car C, it is assumed that it moves to position (2,1) first, then to (1,1) and then moves out of the intersection. Every time any car moves forward it consumes one time step. Once the game has been designed, the payoffs of the three cars can be calculated.

2.1.1 Nine possible payoffs for cars A, B and C

Since there are about eight possible positions in the matrix for a car to exist, excluding (1,1), we start with a game of three cars and calculate their payoffs. We have drawn the game in form of a matrix and have placed the three cars at (3,1), (3,2) and (2,3) and are called A, B and C respectively. When we look at the matrix, positions (2,1) and (2,2) are places exactly in front of these three cars. If we add more cars at these two places, we will be able to make a different game. So for the next game we add a car in either one of these two positions or probably in both of the positions. Since these two positions are common to both the cross lane as well as the main lane, we have two kinds of cars possible. A car moving in the same direction as of A/C or a car moving in the direction of B. Car in the same direction as of A/C will be called a '-1' car and a car moving in the direction of B will be called a '+1' car. So we have three possibilities for each of the two positions, a -1 car, a +1 car or no car. With this arrangement, if we permute the three types of cars at two positions, we have nine games possible for the analysis and calculation of payoffs of cars A, B and C.

We start with a game of three cars placed in (3,1), (3,2) and (2,3) and no car at (2,1) or at (2,2) and call it 'game one'. These three cars are the players for this game trying to cross the road intersection. These players are free to choose their strategy in the game. Each strategy leads to a different payoff for each of the players. Figure 2.2 shows the matrix form of this game. The matrix represents the road intersection where rows 2 and 3 and columns 2 and 3 are the lanes of the intersection. Each square in the matrix is a position block where a car may be placed during the game. The cars are represented with the help of arrows that point in the direction of the car's movement. The two cars at (3,2) and (3,1) are A and C respectively. The third car on (2,3) is car B. Cars A and C together form the cross lane, while car B is part of the main lane.

2.1.2 Payoff structure for the matrix form of game one

The matrix form representation of this game shows three cars standing at a road intersection trying to get across. When the game begins, each car tries to get to a block exactly in front of it. As a car moves from one block to the

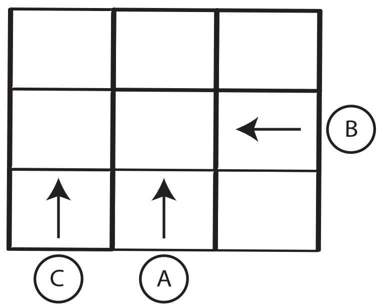


Figure 2.2: Game one represented by a matrix. The arrows A, B and C are three cars of the game that are trying to get out of the matrix. The squares to which these three cars initially belong to are common to all games. Other eight possible games are also represented in this manner with addition of cars at (2,2) or (2,1).

next, it is called one time step. The payoffs of the cars are calculated with respect to these time steps. If a car moves forward, its payoff increases by a factor of 1. For example, in the matrix above, car C has a clear way ahead of it and can get across without any trouble. Hence it gets out of the matrix gaining a payoff of 3 in three time steps.

While trying to get out of the matrix, every car needs to look around at its environment as well. As there could be a situation when a car is present diagonally opposite to it, competing for the same position in the matrix. For example, in this case, it is visible that cars A and B are competing for position (2,2). Whether car A wins or B wins, they both lead to two different payoffs for the two cars. In order to understand better lets say car A wins, it will get across gaining a payoff of 3 in three time steps. Whereas car B is blocked and does not move for the first two time steps. This reduces the payoff of car B to -2 initially because it can start to move only when car A has cleared position (2,2) in the matrix. Once car A is out of the way, car B takes three time steps to get across, increasing its payoff by 3. Hence at the end of this strategy, payoff of the three cars will be [A=3, B=1, C=3].

Second strategy would be if car B wins initially and moves to position (2,2) before A. This increases the payoff of car B to 1 and reduces payoff of car A to -1. But by the the time B reaches (2,2), car C moves to position (2,3) blocking car B's way for the next time step as car B cannot move forward until car C clears out. So by the end of the second time step, payoff of car B reduces back to zero and payoff of car A reduces to -2 as it is still blocked by car B. In the third time step car B clears car A's path and they both start to move out. So the final payoffs for the three cars in this case becomes [A=0,B=2,C=3]

2.2 Extensive form representation

An extensive-form game is a specification of a game in game theory. It shows precise representation of a number of important aspects, like the sequencing of players' possible moves, their choices at every decision point, the information each player has about the other player's moves when he makes a decision, and his payoffs for all possible game outcomes.

The Extensive form of the games is formed in a way that it mostly focuses on the payoff calculation of the cars. As the payoff of the three cars is calculated in time steps, the Extensive form of these games is designed in a way that it shows the payoff after each time step. The following figure shows a part a Extensive form representation of a game. In order to explain the Extensive form representation of the games, we use the very first game of three cars in a matrix competing with each other to get across. The three cars represented by A, B and C are shown in the figure below. If a car moves forward, we represent it by a line segment with positive slope in front of it. While if a car is not able to move forward, it has a line segment with negative slope in front of it. Every positive sloped line segment marks a +1 in that car's payoff while every line segment with negative slope marks a -1 in the car's payoff.

In the first game, car A has two choices, it could either move forward which implies that car B would not be able to move as shown in the up hierarchy in the figure below. Or car A could stay put, giving chance to car B to move forward as shown by the down hierarchy in 2.3. Both hierarchies lead to different payoffs for the cars.

2.2.1 How we calculate payoffs in our games

As we know, in the first game between cars A and B, either A may move forward or car B may move forward leading to two different pathways. The Extensive form representation here shows the two situations as two different hierarchies. Hence, in the above Extensive form representation, there are two lines of both negative as well as positive slopes emerging from car A circle

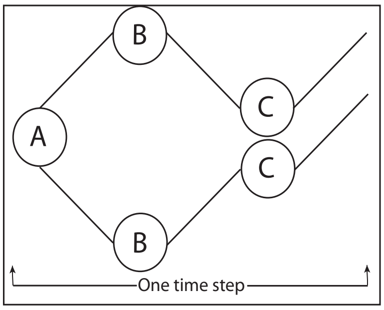


Figure 2.3: Showing a time step in the extensive form of a game. The point from where car A either gains payoff or loses payoff to the end point of the line segment in front of car C is considered as one time step. Change in payoff of the three cars, whether +1 or -1, is considered as one time step. The positive sloped line segment in front of a car shows forward movement of that car. A line segment with negative slope in front of a car shows that the car was not able to move. Therefore the above two different threads of hierarchies occur depending on whether car A wins from B to move forward or car B wins to move forward.

which would lead to two different payoff hierarchies for the three cars. The payoffs calculated after this first time step as shown in the Figure 2.4.

Every time a car moves forward, as shown by the line segment with positive slope, the car gains a payoff of +1. If a car is not able to move either due to another car being present in front of it or due to competition, the car looses payoff by -1. For example, in figure 2.4, in the up path car A has a positive sloped line segment and hence its payoff becomes +1. Whereas car B has a line segment of negative slope, which is because car A won and moved forward, and hence car B has a payoff of -1.

This process of calculation of the payoffs is followed after each time step until each one of the cars has crossed intersection. The head of the long arrow points to the end of a time step in that hierarchy and the tail shows the payoff of the three cars by the end of that time step. One other situation would be when a car is out of the intersection while there are other cars still in side the matrix, a line segment with zero slope is placed in front of that car's circle in the extensive form. This line segment shows that the payoff of that car freezes for the rest of the time steps.

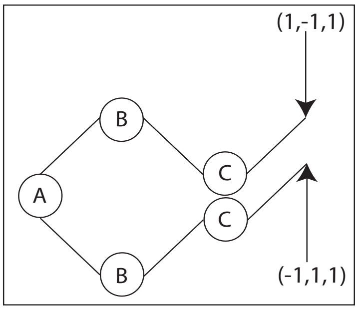


Figure 2.4: Extensive form of a game showing payoffs for the three cars A, B and C after first time step in the game. Two different payoffs occur depending on whether car A wins or looses. The positive sloped line in front of a car shows a +1 gain in payoff. While the other line segment with negative slope marks a loss of -1 in the payoff of a car.

2.2.2 Calculation of payoffs in game one using extensive form

In the Extensive form, when there are two or more threads of payoffs possible, in order to calculate the final payoff of the cars, we take the average of the two payoff threads. As the following figure 2.5 shows a game in its Extensive form, it is the most simple game with three cars inside an intersection. As discussed above, there are two hierarchies of payoffs possible. The first branch originates when A decides to move first while the second branch is formed when B wins and moves forward. Since there is no car to interfere with C's path, payoff of C remains same at the end of both hierarchies. The two payoffs received at the end of the hierarchies are [3,1,3] in case 1 and [0,2,3] in case 2. In order to find the final payoff, we take the average of the two payoffs. Hence, the final payoff for this game becomes [1.5,1.5,3].

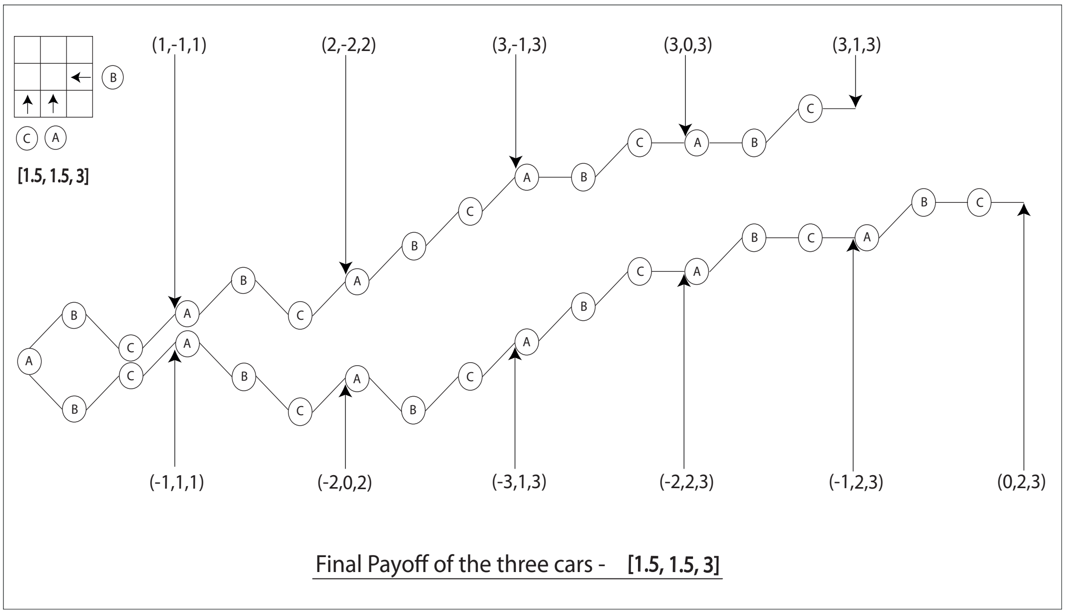


Figure 2.5: Extensive form representation of a game with three cars A, B and C placed as shown in the matrix. The extensive form shows how payoffs are calculated by hand. The payoffs of the three cars have been shown after each time step in both the possible hierarchies. The final payoff is obtained after taking average of the two payoffs at the end of the two hierarchies.

Chapter 3

Toy games to analyze an extended version of the standard snow drift game

We have tried to modify this standard snow drift game into a game where one of the players has a partner. And then we look for possibilities of cooperation among them. We have designed nine games to understand the decisions made at traffic intersections better. In these designs, we have made a traffic intersection which is similar to the intersection near Symantec building on Baner Road. The matrix consists of 3 cars coming from 2 different lanes and competing at the intersection. The three cars are placed at same places as described in the matrix form. Cars A and C form one lane of the intersection and car B forms a different lane, therefore these three cars are our players we find the payoff for. Any other car being added in the matrix will only act as hindrance in the path of these three cars.

The other eight games are designed if we increase the number of cars in the second row of the first game, it would result into a different game with different payoffs for the three cars. Even the direction of the car being added to the game would matter and will result into a completely different game. For example, if we add a car at (2,2), whether it is a car moving towards North or West will give two different games. For simplicity we shall call a North moving car as -1 car and a West moving car a +1 car. So if there is a -1 car at (2,2) in the same previous game, it will move out of the game in first time step itself. But if there is a -1 car at (2,2), it will compete with C and reduce C's payoff in the process.

In all the games, initial structure is - A is placed in position (1,2), B in position (2,1) and C is in position (1,3). Each of the 9 games is different from the other as positions (2,2) and (2,3) may or may not be filled with cars that form the traffic. For each of the two positions (2,2) and (2,3) there are 3 choices, a -1 car, a +1 car or no car at all. By permutation of the 3 type of cars with 2 places for them, it is clear that nine games are possible with only slight changes.

3.1 Payoff structure

As we know every time a car moves forward, whether the other 2 cars have moved or not, it is considered as 1 time step. For each time step, a forward movement of a car is considered a +1 in its payoff and each time a car is not able to move due to traffic, it is considered as a -1 in its payoff.

The overall payoff of a car for a game is calculated by taking the average of the situations for each car. In order to understand the calculation of payoffs for the 9 cases, extensive form for each one of the games have been drawn.

The extensive form of the games is like a hierarchy table showing the calculation of the payoffs at every time step. In the table, the lines with positive slope show that the car moves forward and gains a +1 in its payoff. The lines with negative slope show that the car is made to wait at its position for that time step and gets a -1 in its payoff. The straight line shows that the car has crossed the intersection and its payoff freezes to what it was at the start of that time step. The start of the arrows in the hierarchy tables show the payoff at that time step, while the end of the arrow point to which time step it speaks of.

3.1.1 Game one

This is the simplest of all games with cars A, B and C in the matrix and no other car anywhere else in the matrix. Cars A and B at positions (3,2) and (2,3) respectively, start to compete for position (2,2). In the first time step, one of them will win while car C will move to (2,1). If A wins, it moves the next steps forward along with C making their payoffs equal in this case as shown by the upward hierarchy in the table. Payoff of B becomes -1 as it has to wait for A to clear the road. In the second time step as well B remains in its position reducing its payoff to -2 while A and C move to (1,2) and (1,3) respectively. Now that B has a clear path in front of it, it moves to (2,2) increasing its payoff to -1. By the end of third time step cars A and C are out of the game with payoff of 3 each and B is at (2,2). B takes two more time steps to cross the intersection, and comes out of the game with a final payoff of 1. At the end of this hierarchy, payoff of the three cars is [A=3, B=1, C=3].

Another situation would be if B wins the competition with A, B would move to (2,2) in the first time step with a payoff of 1 and making A's payoff -1. By this time C would have moved to (2,1) which would block B's pathway for the next time step. So after second time step, C will be at (1,1), B will be at (2,2) with payoff back to zero and A will be at (3,2) with -2 payoff. Now B moves to (2,1) in the third time step, clearing way for A. A starts to move in the fourth time step with -3 payoff while B moves out of the game with payoff of 2. A gains a payoff of 3 by the end of six time steps but overall its payoff becomes zero. So the final payoff of this hierarchy is [A=0, B=2, C=3].

The final payoff of the game is calculated by taking average of the two hierarchies. Here the payoff of the game becomes [A=1.5, B=1.5, C=3]. Payoff of A and B are same which is worth noticing, as the payoff implies that the two cars exist in symmetric situations.

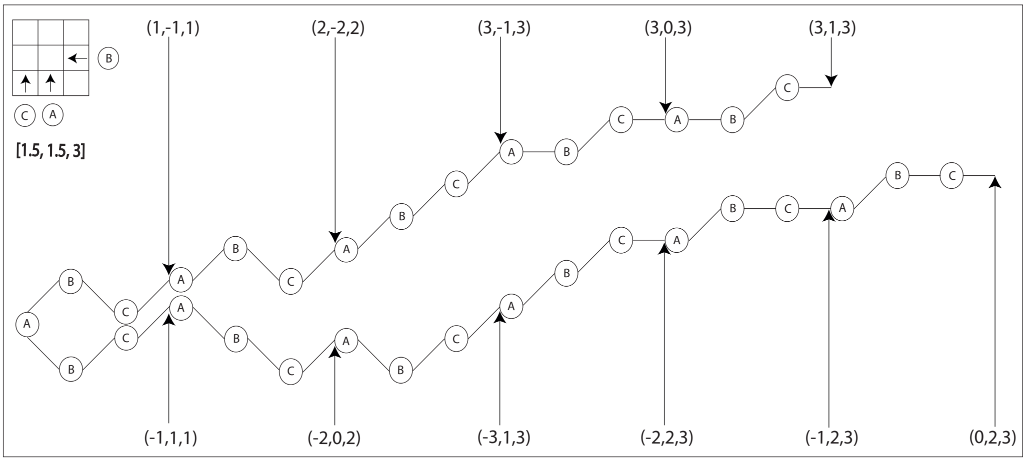


Figure 3.1: Extensive form of game one

3.1.2 Game two

In the second Game, we introduce a -1 car at position (2,1), the rest of the design remains the same with A, B and C at (3,2), (2,3), and (3,1) respectively. This -1 car becomes a hindrance for C for the first time step, reducing payoff of C to -1. A and B compete for position (2,2) and their result leads to two different payoff for the cars. If A wins, then after first time step, it moves to (2,2) gaining a payoff of 1, B looses payoff by 1 and C also looses payoff because of the -1 car. In the next time step, a clears the way for B and C starts to move as well. By the end, A gains a payoff of 3 and payoff of C becomes 2. Payoff of car B becomes 1 as A takes two time steps to clear the way for B. The payoff at the end is [A=3, B=1, C=2]

The other situation would be if B wins initially. Then in one time step, B would move to (2,2) and since the -1 car will also take one time step to clear the way, C will remain at (3,1). Here arises a situation where B and C will have to compete for the position (2,1), this will further result into two branches. First one is if B wins again, once B reaches position (2,1) path for A will clear up and it will start to move although by this time A has a payoff of -2. While C would not be able to move for another time step which reduces its payoff to -3. Hence at the end of this branch, payoff of the three cars will be [A=1, B=3, C=0].

Now if after the first time step, C had decided to move to (2,1), B would not have been able to move for two time steps which makes its payoff -1. Car A will remain at its position for a total of four time steps which makes its payoff -1. Hence the payoffs for this branch would be [A=-1, B=1, C=2]. The final payoff is calculated by taking average of the two split 2nd and 3rd branches first, i.e. [A=0, B=2, C=1], and then taking average of this payoff with the payoff from the first branch. So the final payoff of the game becomes, [A=1.5, B=1.5, C=1.5]

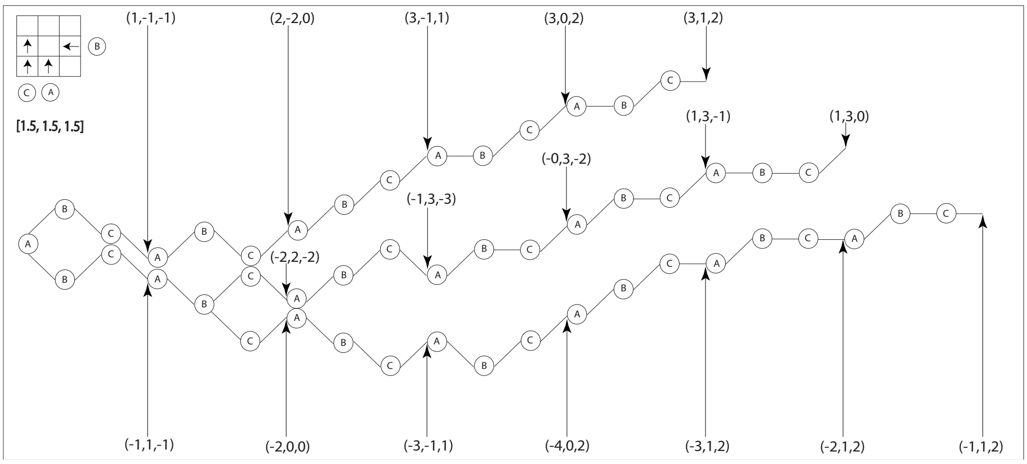


Figure 3.2: Extensive form of game two

3.1.3 Game three

In Game Three, we have a 3x3 matrix of 4 cars and the structure is similar to that of Game Two. The only difference is that the new car that has been introduced at position (1,2) is +1 car, it is moving in the same direction as car B does. This +1 car clears the path for car3 in one time step exactly and hence the rest of the calculation of payoffs for the three cars becomes the same as in Game two.

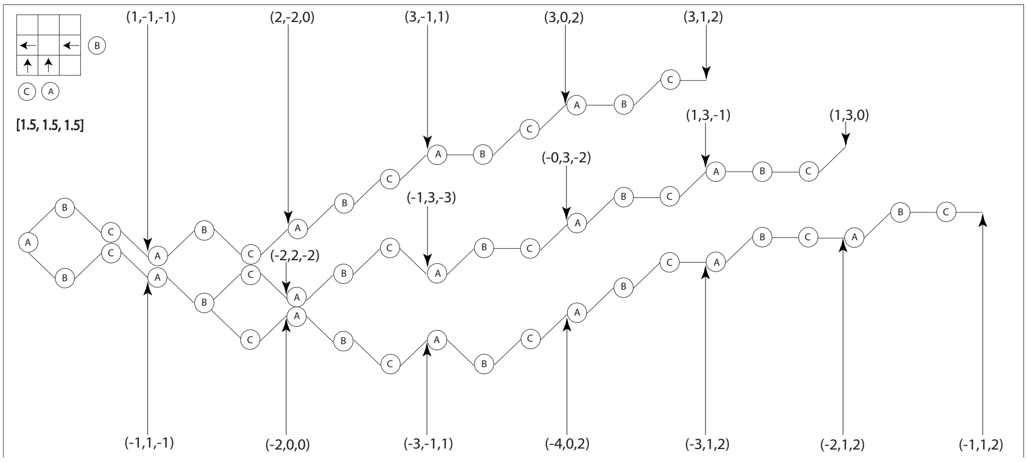


Figure 3.3: Extensive form of game three

3.1.4 Game Four

The fourth game is a simple game of four cars, with the usual A, B and C cars at (3,2), (2,3) and (3,1) and a -1 car is placed at position (2,2). Since it is a -1 car, it will be out of the game in first time step itself as it will move to (1,2) and will no longer hinder any other car 's way. C will easily move out of the intersection three time steps with a payoff of 3. Payoff of cars A and B will be reduced to -1 each because of the -1 car. In the second time step, one of them will move forward. If B moves forward, it will move out of A's way in two time steps reducing payoff of A to -3. A starts to move in the fourth time step and B moves out of the intersection after the fourth time step. Payoff of the three cars in this hierarchy becomes [A=0, B=2, C=3].

If A wins then payoff of B will be reduced to -3 before it can even start to move. In this case A and B are in symmetric situations. Payoff of this hierarchy becomes [A=2, B=0, C=3]. The final average payoff of the game becomes [A=1, B=1, C=3].

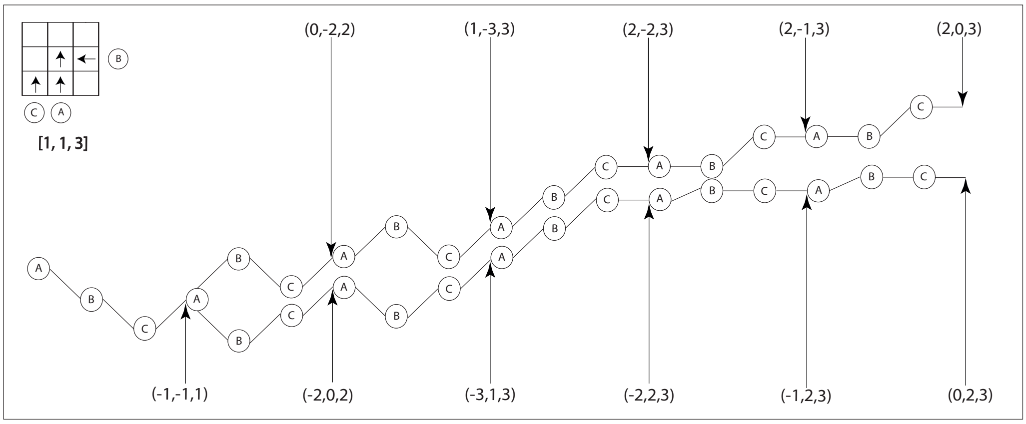


Figure 3.4: Extensive form of game four

3.1.5 Game five

In Game Five, in the 3x3 matrix we have introduced 2 more cars at positions (2,1) and (2,2). The car at (2,1) is a +1 car and the car at (2,2) is -1 car. These cars work as hindrances in the path of A, B and C. Although, when we notice the car at (2,1) will move out of the game in 1 time step as it is at the edge of the intersection. While the -1 car at (2,2) will also move out of the game after the first time step as soon as it moves to (1,2). In one time step, these two cars clear the path for the three cars in study leaving behind a structure similar to Game One. Although these two cars reduce the payoffs of the three cars by 1. Once they are out, cars A and B start to compete for position (2,2) while C moves forward unhindered. The branch splits into two as in the first game. The final payoff for the cars in this case becomes [A=0.5, B=0.5, C=2].

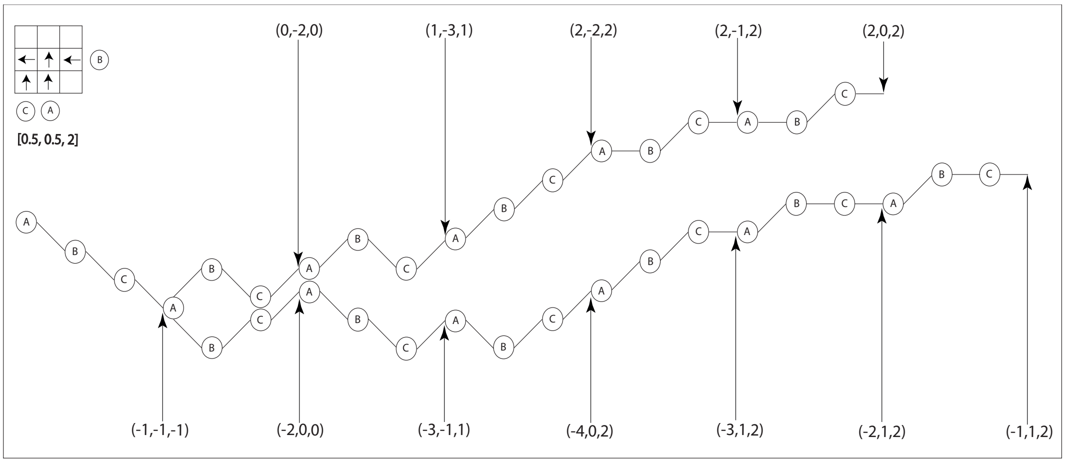


Figure 3.5: Extensive form of game five

3.1.6 Game six

Game Six consists of a 3*3 matrix with 5 cars. The cars at position(2,1) and (2,2) are -1 cars, that is they both are moving in the same direction as car C. The two -1 cars will clear out of the game in one time step. As soon as they are out, the structure will become the same as that of game five. Cars A and B will compete for (2,2) after the first time step and this will lead to two different payoffs for the cars. The calculation is same as that of game five. The final payoff is also the same which is (A=0.5, B=0.5, C=2).

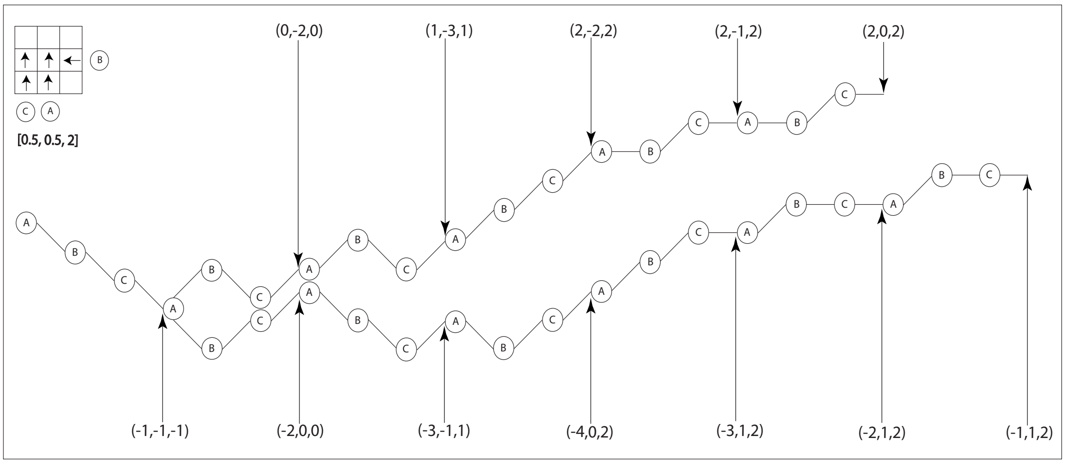


Figure 3.6: Extensive form of game six

3.1.7 Game seven

Game seven is a game with four cars. The fourth car is placed at (2,2) and it is a +1 car. Which means this +1 car will be competing with C for position (2,1). Here itself the hierarchy divides into two branches, one if car C moves forward to (2,1) and the other if the +1 car moves to (2,1). If C moves first, its payoff will be 3 for that branch. While cars A and B will loose payoff for the first three time steps, as C will clear the way in two time steps and +1 car will clear A and B 's way in one more time step. After this, A and B will engage into a competition for position (2,2) which further divides the hierarchy into two. Since both their payoff are already -3 each, the one that looses in the competition for (2,2) will loose a payoff of -2 more and the other will gain a payoff of 2. The end payoffs of these two branches will be (A=0, B=-2, C=3) if A wins and (A=-2, B=0, C=3) if B wins. and their average is (A=-1, B=-1, C=3).

If at the start the +1 car moves forward to (2,1), payoff of car C will be reduced to -2 in the first two time steps. By the end of the first two time steps, either of cars A and B would have moved forward. If A wins, B stays put for two more time steps and starts to move with a payoff of -3 and comes out of the intersection with a payoff of 0. C will come out of the game with a payoff of 1. While A started with a payoff of -1 and will come out with a payoff of 2. The payoff at the end of this branch is (A=2, B=0, C=1).

Another branch is if B wins, then after the second time step B will be at position (2,2) competing with car C for position (2,1). This will further lead to branching of the hierarchy. In the competition with B, if C looses, it will have to wait for two more time steps to start which will make its payoff -1. As for B, it will move out of the game with payoff 2. As soon as B moves to (2,1), A starts to move and hence gets a payoff of 0. The final payoff of the three are (A=0, B=2, C=-1).

Now in the competition with B, if C wins, it will come out of the intersection with payoff 1. Payoff of B will be reduced by -2 and its payoff at the end will be 0. Payoff of A is highly reduced as B is blocking its way for 5 time steps and hence the payoff of A at the end is -2. Payoff of all three cars is (A=-2, B=0, C=1). Taking the average of the last two payoffs gives (A=-1, B=1, C=0), taking average of this and the third branch gives (A=0.5, B=0.5, C=0.5). Now taking average of this quantity and the first average we took gives the final payoff for the game as (A=-0.25, B=-0.25, C=1.75).

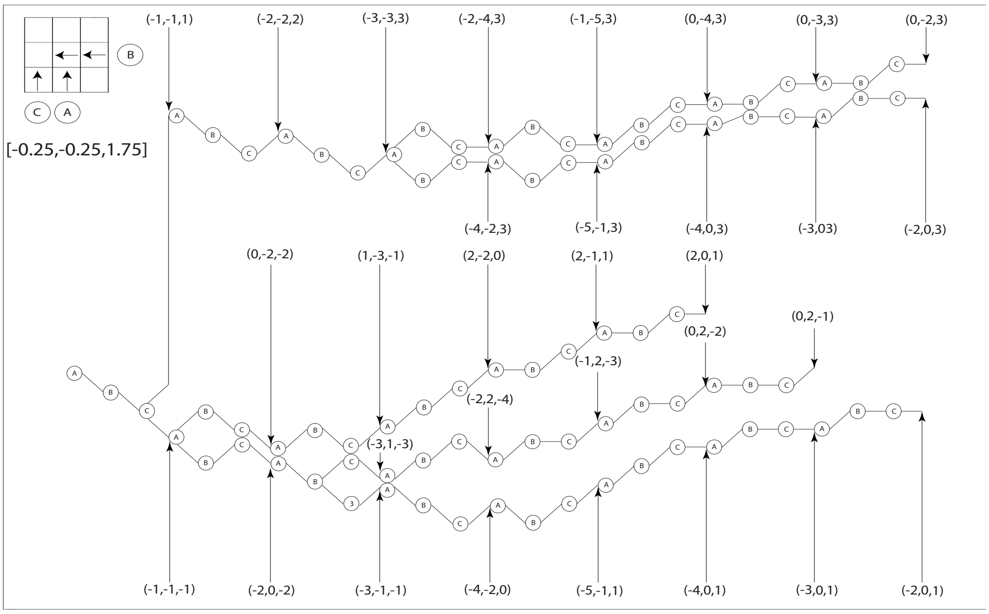


Figure 3.7: Extensive form of game seven

3.1.8 Game eight

As it is clear from the matrix, that +1 car at position (2,1) will consume one time step in moving out, hence reducing the payoffs of all 3 cars to -1. In the second time step, car3 and +1 car at position (2,2) compete for position (2,1), which diverges the hierarchy into two. If C wins and moves across in 2 time steps, A and B would compete later for position (2,2) further diverging the hierarchy into two branches. But if the +1 car at position (2,2) wins from C, and if B wins from A, then B and C compete later for position (2,1). Hence the hierarchy in this game diverges into five different branches. The payoffs of A and B reduces to negative numbers, and payoff of C reduces but still remains at least positive, which is expected because C is not as crowded as A and B. The payoff calculation of game eight becomes the same as that of game seven after the first time step. The only difference between the two is the +1 car at (2,1) which reduces the payoff of the three cars by -1 at the start. And hence the final payoff for the cars in game eight is one less than that in game seven. The final payoff is (A=-1.25, B=-1.25, C=0.75).

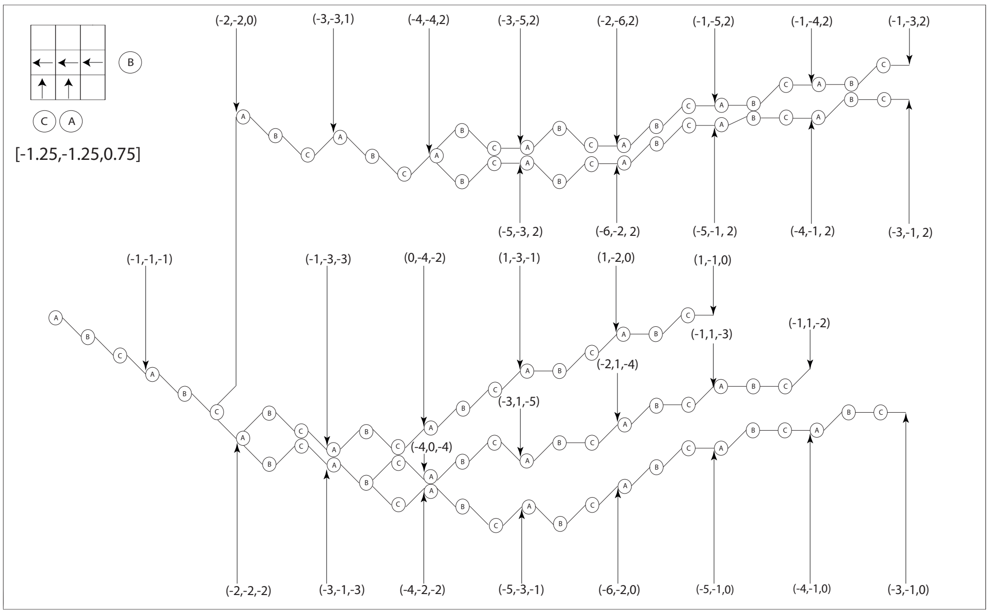


Figure 3.8: Extensive form of game eight

3.1.9 Game nine

Game Nine consists of five cars in a 3x3 matrix. Cars A, B and C and two other cars are added at (2,1) and (2,2). Car at (2,1) is a -1 car and the car at (2,2) is +1 car. The car at (2,1) will consume one time step to move out of the matrix, making the payoffs of the three cars equal to -1. After the first time step, the +1 car from (2,2) competes with C. Depending on whether C wins or looses the branch diverges into two. Now if C looses, then by the time the +1 car moves out of the matrix, cars A and B compete for position (2,2). Now whether if B win and moves to (2,2) it will have to compete with C for position (2,1). Hence the whole extensive form representation of this game is divided into five branches. The calculation of the payoffs in this game is same as that of game eight as the only difference between the two is the direction of the car at position (2,1) initially which moves out of the game in one time step affecting both the games in the same manner. Therefore the payoff of the three cars in this game is also (A=-1.25, B=-1.25, C=0.75).

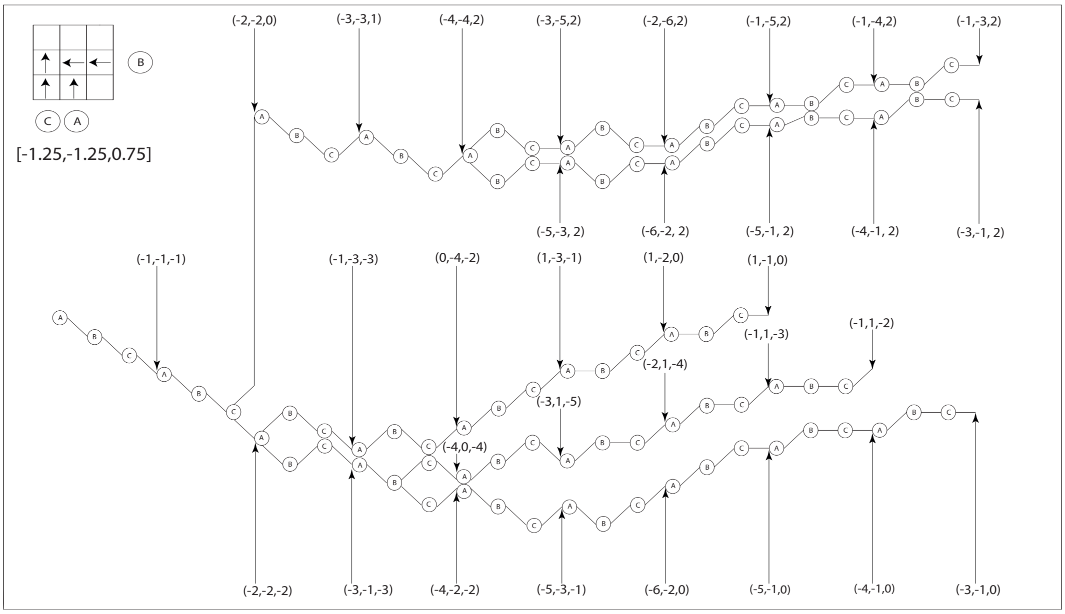


Figure 3.9: Extensive form of game nine

Chapter 4

Confirmation of payoffs calculated using extensive form

Since the payoff calculation using extensive form has been done by hand. In order to confirm that this payoff calculation method is correct, payoffs of the three cars have also been calculated by writing matlab algorithms in each of the nine games. In the extensive form, each game results in a number of possible payoffs. Like in game one with only cars A, B and C playing, the game has two possible payoffs (A=3, B=1, C=3) and (A=0, B=2, C=3). Therefore the codes are run for a large number of times so that the average gives accurate results. The code for game one is added below.

4.1 Matlab Code

In order to relate the matrix with the intersection, in the algorithm the cars inside a matrix 'T' are shown either by a +1 or a -1. A +1 car is the one moving towards West, that is the direction of car B, and the -1 car is the one moving towards North, which is the direction of cars A and C. Another matrix 'Tnew' is created at the start of the code, which has entries equal to T itself. Whenever a car moves in the code, the entries are first stored in the matrix Tnew because we need to check all positions of the matrix in one time step without changing them until the end of that time step. The payoff calculation in the code is similar to the ones done by hand, whenever a car moves forward, it gains a payoff of +1 and when a car is not able to move it looses payoff by -1.

The code starts from position (1,2), which always has a -1 car, it checks whether the position in front of this car is free or not. If it is free, that is if position (2,2) is unoccupied, then the code checks if there is another car competing for the same available spot. It checks if there is a car at (2,1) or not. If there is no car at (2,1), car at (1,2) can simply move forward. When a car moves forward, the position it left behind changes to 0 while the position it moved into becomes equal to the number of the car in the matrix Tnew. Here (1,2) in the matrix T becomes 0 and the position (2,2) becomes -1. But since initially there is always a car at (2,1). The two cars from (1,2) and (2,1) compete for position (2,2). The code is generates a random number between 0 and 1, every time two cars in the matrix face competition. Based on the value of the random number, the code decides whether a +1 car moves forward or a -1 car moves forward. If the random number generated is greater than 0.5, +1 car moves to the next position. But if the random number generated is less than 0.5, -1 car moves to the next empty position.

A matrix S with all elements equal to 1, has been created in the code. This matrix keeps track of all those positions which have already been dealt with and there is no need to check them at later stage. For example, while dealing with the competition between cars from position (1,2) and (2,1), if the code decides that car from position (1,2) moves forward, then the code does not need to come back to position (1,2) anymore. So the code creates a 0 at position (1,2) in the matrix S. If in another time step, the code comes back to checking position (1,2) of T, it will first have to check whether position (1,2) of S is 0 or +1. If it is 0, the code will move to the next position in T. But if it is +1, the code will start to check the environment of that position for various possible moves.

Once the code has taken care of position (1,2) of T, it moves to position (1,3) and checks whether position (2,3) is available for movement or not. If (2,3) is unoccupied, then the code looks for a car at (2,2). If (2,2) is unoccupied too, car from (1,3) moves forward gaining a payoff of +1. If (2,2) is occupied the code checks whether it is a +1 car or a -1 car at (2,2). If it is a -1 car, the car from (1,3) can still move forward but if it is a +1 car at (2,2) then there is competition between the two cars and a random number generated by the code determines which will move forward. The code goes through every position of the matrix T in one time step, following the same process. A time step ends when the code is done with position (2,3).

After each time step, the code changes the matrix 'T' to matrix 'Tnew' with all the required changes that were made in Tnew during that time step. This is done, so that for the next time step the code checks the environment of cars from T while it makes changes to Tnew. P, Q and R are representations for payoffs of cars A, B and C which are equal to zero initially. The values of P, Q and R changes after each time step depending on whether a particular car was able to move forward or not. Once all the payoff have been recorded, the code takes average of the payoffs over the 10000 runs and stores the final payoffs in E, F and G. These payoffs for all three cars in case of each of the nine games is shown in a payoff table in figure 4.1.

```
K=0;
L=0;
for i=0:9999
P=0;
Q=0;
R=0;
T = [0 -1 -1; 1 0 0];    %Initial matrix T
 t=0;                    %program starts at time=0
 Tnew=T;
 for t=0:8
     S = [1 1 1; 1 1 1];%2*3 matrix of all 1's, which changes when a
                        %particular position is dealt with
     t=t+1;

%===========================================

%Evaluating position (1,2)----
%===========================================

  if T(1,2)~=0     %Checks if there is a car at (1,2) or not
      if S(1,2)~=0    %If yes, looks if it has been dealt with or not
          if T(2,2)==0   %Looks for a vacancy in front of (1,2)car, i.e.
                         %at position (2,2)
              if T(2,1)~=0  %Checks if there is any competition for the spot (2,2)
                  compete_rand=rand(1,1);
                  if compete_rand<0.5      %If yes, -1 car from (1,2) moves
                                           %forward for compete_rand<0.5
                      Tnew(1,2)=0;
                      Tnew(2,2)=-1;
                      S(1,2)=0;
                      S(2,1)=0;
                      P=P+1;
                      Q=Q-1;
                  else         %And 1 car from (2,1) moves forward
                               %for compete_rand>0.5
                      Tnew(2,1)=0;
                      Tnew(2,2)=1;
                      S(2,1)=0;
                      S(1,2)=0;
```

```
                P=P-1;
                Q=Q+1;
            end
        else          %If there isn't any competition,
                      %then -1 car from (1,2) moves to (2,2)
            Tnew(1,2)=0;
            Tnew(2,2)=-1;
            S(1,2)=0;
            P=P+1;
        end
    else
        P=P-1;
    end
  end
end

%============================================

%Evaluating position (1,3)------
%============================================


  if T(1,3)~=0     %Checks if there is a car at (1,3)
      if S(1,3)~=0    %Confirms if the car has been dealt with or not
          if T(2,3)==0   %Looks if there is a car in front of (1,3), i.e. at (2,3)
              if T(2,2)==1  %Checks if there is a car at (2,2) competing for the (2
                  compete_rand=rand(1,1);
                  if compete_rand<0.5     %If yes, for compete_rand<0.5,
                                          %-1 car from (1,3) moves to (2,3)
                      Tnew(1,3)=0;
                      Tnew(2,3)=-1;
                      S(1,3)=0;
                      S(2,2)=0;
                      if T(2,1)==0
                          Q=Q-1;
                      end
                      R=R+1;
                  else          %Otherwise, for compete_rand>0.5,
                                %car from (2,2) moves to (2,3)
                      Tnew(2,2)=0;
                      Tnew(2,3)=1;
                      S(1,3)=0;
                      S(2,2)=0;
                      if T(2,1)==0
                          Q=Q+1;
                      end
                      R=R-1;
                  end
              else          %If there is no car competing,
```

```
                    %car from (1,3) moves forward to (2,3)
               Tnew(1,3)=0;
               Tnew(2,3)=-1;
               S(1,3)=0;
               R=R+1;
           end
       else
           R=R-1;
       end
   end
end

%===========================================

%Evaluating position (2,1)-----
%===========================================

if T(2,1)~=0     %Checks if there is a car at (2,1)
    if S(2,1)~=0    %Confirms if the car has been dealt with or not
        if T(2,2)==0   %Looks if there is a car in front of (2,1), i.e. at (2,2)
            if T(1,2)==-1 %Checks if there is a car at (1,2)
                          %competing for the (2,2) spot
                if S(1,2)~=0
                    compete_rand=rand(1,1);
                    if compete_rand<0.5     %If yes, for compete_rand<0.5,
                                            %-1 car from (1,2) moves to (2,2)
                        Tnew(1,2)=0;
                        Tnew(2,2)=-1;
                        S(1,2)=0;
                        S(2,1)=0;
                    else          %Otherwise , for compete_rand>0.5,
                                  %car from (2,1) moves to (2,2)
                        Tnew(2,1)=0;
                        Tnew(2,2)=1;
                        S(2,1)=0;
                        S(1,2)=0;
                    end
                end
            else          %If there is no car competing,
                          %car from (2,1) moves forward to (2,2)
                Tnew(2,1)=0;
                Tnew(2,2)=1;
                S(2,1)=0;
                Q=Q+1;
            end
        else
            Q=Q-1;
        end
    end
```

```
  end

%==========================================

%Evaluating position (2,2) for +1 car------
%==========================================

  if T(2,2)==1      %Checks if there is a +1 car at (2,2)
      if S(2,2)~=0     %Confirms if the car has been dealt with or not
          if T(2,3)==0    %Looks if there is a car in front of (2,2), i.e. at (2,3)
              if T(1,3)~=0  %Checks if there is a car at (1,3)
                            %competing for the (2,3) spot
                  if S(1,3)~=0
                      compete_rand=rand(1,1);
                      if compete_rand<0.5      %If yes, for compete_rand<0.5,
                                               %-1 car from (1,3) moves to (2,3)
                          Tnew(1,3)=0;
                          Tnew(2,3)=-1;
                          S(1,3)=0;
                          S(2,2)=0;
                      else          %Otherwise , for compete_rand>0.5,
                                    %car from (2,2) moves to (2,3)
                          Tnew(2,2)=0;
                          Tnew(2,3)=1;
                          S(1,3)=0;
                          S(2,2)=0;
                      end
                  end
              else          %If there is no car competing,
                            %car from (2,2) moves forward to (2,3)
                  Tnew(2,2)=0;
                  Tnew(2,3)=1;
                  S(2,2)=0;
                  if T(2,1)==0
                      Q=Q+1;
                  end
              end
          else
              if T(2,1)==0
                  Q=Q-1;
              end
          end
      end
  end

%==========================================

%Evaluating position (2,2) for -1 car------
%==========================================
```

```
  if T(2,2)==-1   %Checks if there is a -1 car at (2,2)
      Tnew(2,2)=0;
      if T(1,2)==0
          P=P+2;
      end
  end

%===========================================

%Evaluating position (2,3) -----
%===========================================

  if T(2,3)==-1   %Checks if there is a -1 car at (2,3)
      Tnew(2,3)=0;
      S(2,3)=0;
      if T(1,3)==0
          R=R+2;
      end
  end
  if T(2,3)==1
      Tnew(2,3)=0;
      S(2,3)=0;
      if T(2,1)==0
          Q=Q+1;
      end
  end
  T=Tnew;
  S;
 end
 if P==3
     K=K+1;
 elseif P==0
     L=L+1;
 end
end
E=((3*K)+(0*L))/10000
F=(K+(2*L))/10000
G=((3*K)+(3*L))/10000
```

4.2 Payoff table

Payoffs for all the three cars in the nine games are placed together in a table in figure 4.1. The table shows the cumulative payoffs calculated by the codes which are run for 10000 times. Another table, figure 4.2, shows payoffs of the three cars in each of the nine games when calculated using extensive form. These payoff tables help to compare and analyze the payoffs of the three cars. Each game has at least a minimum of two possible payoffs as shown in the extensive form of the games. The matlab code for any game, if run for a fewer number of times may result in any one of the many possible payoffs for that game. But the average taken over the 10000 runs of a code results into a payoff which is almost equal to the payoff received from calculating a final average payoff in the extensive form of that game. For example, in the first game payoffs of car A and B calculated using matlab are 1.5156 and 1.4948 respectively. While payoffs of cars A and B calculated using extensive form is 1.5 for both of them, which is almost same as the ones calculated using matlab. Since the two calculation of payoff methods yield equal results, this confirms that the extensive form used to calculate the payoffs is correct.

From table 4.2, we can observe that the payoffs of car A and car B are always same in each of the nine games. This shows that A and B exist in almost symmetric situation when compared to each other and that is why their payoffs are always same in all the nine games. While payoff of C is always higher than those of A and B. Which is expected, since C is always less crowded than the other two cars.

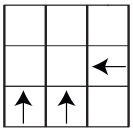
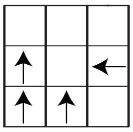
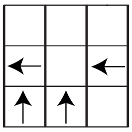
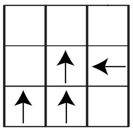
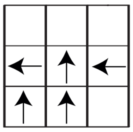
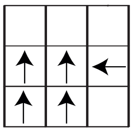
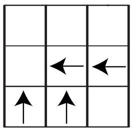
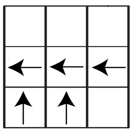
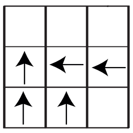
	Payoffs→ / Matrix↓	Car A	Car B	Car C
1.		1.5156	1.4948	3
2.		1.4987	1.4957	1.5043
3.		1.4991	1.4999	1.5001
4.		1.004	0.9960	3
5.		0.4961	0.5013	2
6.		0.5126	0.4958	2
7.		-0.2578	-0.2502	1.7472
8.		-1.2678	-1.2418	0.7380
9.		-1.2556	-1.2528	0.7616

Figure 4.1: Payoffs of cars A, B and C in each of the nine games when calculated using matlab. These payoffs are compared with payoffs in the next table. The payoffs here confirms that the payoffs calculated by extensive form are correct.

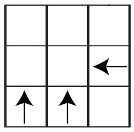
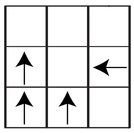
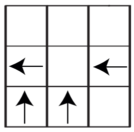
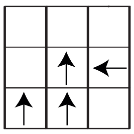
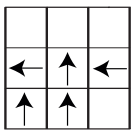
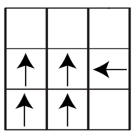
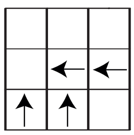
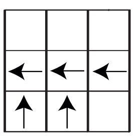
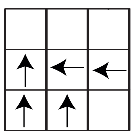
	Payoffs→ / Matrix↓	Car A	Car B	Car C
1.		1.5	1.5	3
2.		1.5	1.5	1.5
3.		1.5	1.5	1.5
4.		1	1	3
5.		0.5	0.5	2
6.		0.5	0.5	2
7.		-0.25	-0.25	1.75
8.		-1.25	-1.25	0.75
9.		-1.25	-1.25	0.75

Figure 4.2: Payoffs of the three cars in each of the nine games calculated using the extensive form of the games. Payoffs calculated using matlab in table 4.1 result in almost same payoffs as in this table. The games are written in decreasing value of payoffs of A and B.

Chapter 5

Insights into spontaneous lane crossing in the absence of enforcement

We started our problem with the observed high level of cooperation at traffic intersections where there are no traffic lights or traffic policeman. We even recorded a lot of videos of such a intersection near the Mahabaleshwar Hotel on baner road, Pune. While going through the payoffs for the games and the videos that we have of the intersection, it comes to notice that people are gaining results out of conflict. And this conflict between the two lanes has lead to cooperation between the vehicles of the cross lane. If people were to defect, and we expect everyone to be selfish then these intersections would always result into a traffic jam situation. But we also know that is not the case every time, even at the crowded intersection such as the Mahabaleshwar hotel, there is almost never a traffic jam.

5.1 Lane encroachment

Since this conflict is leading cars to cooperate with others, we study various images from the videos of the intersection. These images show various ways of crossing an intersection. In all these images we have tried to observe a pattern.

The first one shows a truck bullying its way through the crowd. It has brought all the traffic in the next lane to a halt and it is crossing the road on its own. The next image is of a single lone rider who plans on crossing the road on its own and in the video he succeeds in doing so. The third one shows a bunch of vehicles crossing the road as a herd. They seem to have blocked the other lane in unison. The fourth image has a white car crossing the road by taking advantage of the truck and there is also a yellow car that takes advantage of the white car to cross the road. These two are perfect examples of parasites.

Figure 5.1: The red truck is an example of how a bigger vehicle can encroach its way through the main lane to get across.

In all these examples, there is one thing that is common, all of these ways of crossing the road shows cooperation among the people like players in a game, they do not talk it out and decide what to do but still they seem to be cooperating. Which brings us to the question, why are people cooperating with each other? In every image, whether it be a bully, a lone rider or the herd, they are all using the same strategy of lane encroachment. The vehicles trying to cross, choke the first lane by encroaching into the road, then they block the second lane and that is how they finally move out of the intersection. Which proves that in every example people are gaining result out of conflict.

The last image figure 5.5 shows the perfect example of lane encroachment. The four continuous images are from the Mahabaleshwar hotel intersection. We can see in the first image, the bunch of vehicles are waiting to encroach

Figure 5.2: The lone rider on the bike from the cross lane is trying to cross through the main lane on its own.

Figure 5.3: This image shows a bunch of vehicles from the cross lane crossing the intersection as a 'herd'. Since it is not very easy for a single vehicle to cross a crowded road intersection.

Figure 5.4: The white car from the cross lane acts as a 'parasite' over the truck. It takes advantage of the truck and crosses the lane along with the truck. While the yellow car to the left of the white car acts as a parasite over the white car in crossing the intersection.

into the lane. In the second image they start to encroach into the lane by blocking the first row of vehicle from the competing lane. In the third image they move forward like a herd choking the second lane as well. And in the fourth image they are out of the intersection. This also shows the matrix design for all the games we have used. The bunch of vehicles waiting are cars A and C while in the other lane, first lane is car B.

Figure 5.5: The four images are continuous clicks of a bunch of vehicles from the cross lane trying to cross through the main lane. The first image shows how vehicles are accumulating to form a 'herd'. In the second image, the herd encroaches into the first row of vehicles of the main lane. The third image shows the herd encroaching into the second row of vehicles in the main lane. In the last image, they are out of the intersection.

5.2 Comparative analysis of a game between A and C

While analyzing the above games for various theories and considering the intersection near Mahabaleshwar hotel, one characteristic theory that becomes visible is of Lane Encroachment. One of the very common observations in the games above is that cars A and B always have the same payoffs which shows that they exist in symmetric situation. Another would be that car C always has a higher payoff than A and B. While going through the observations, we start to notice whether A and C are dependent on each other or not.

For the sake of this observation, we have designed another game where we compare the payoffs of cars A and C. This game is played with each of the nine designs of the matrices that we used in the earlier games. In figure 5.6, we have a table showing six cases of the nine games that were played earlier. We have merged the nine games into six cases as there were a few games that had similar results. For example, game two and three result into same payoffs for the three cars, the only difference between the two is the direction of the car at position (2,1). This car affects the payoff of the three cars in both games in the same manner since it moves out of the matrix in one time step. Therefore game two and game three are being shown as case 2 in this table. Similarly, games five and six and games eight and nine have equal payoffs too and that is why they have been merged into case 4 and case 6 respectively.

In figure 5.7, we have another table that shows the structure for this new game. Since we are trying to relate cars A and C, if we were to remove one of these two cars from the game and calculate the payoff of the other cars we would be able to look at the relation between the two cars. We start by removing car C from each of the nine games and then calculating the payoff of the cars left in the matrix. Next, we will remove car A from the game and calculate payoff of cars B and C. We know the payoff of car A in presence of car C from the games played earlier, after this game we shall have the payoff of car A in absence of car C as well in each of the nine games. We will also have the results for car C as well in both presence and absence of car A. Once we have these, we can compare the payoffs and look for dependency of the two cars on each other.

In this table we have placed the six cases into three different columns with three different matrices. In the first column we have matrices of the nine possible games. The first case of this game shows game one that is played with three cars. The first column of case 1 has a matrix with cars A, B and C placed at their initial positions. In the second column, we have removed car C while A and B are still at their initial position competing for position (2,2) of the matrix. In the third column, we have removed car A from the original game and allowed cars B and C to get across. The payoffs for the cars have been calculated using extensive form and are written in the next three columns for each of the three cars.

In the second column of the case 1, we have cars A and B competing to get across, calculation of payoff for this case will give two hierarchies in the extensive form. If A wins, car B loses payoff for two consecutive time steps because car A will take two time steps to clear B's way and the payoff becomes [A=3, B=1]. While if B wins, car B will consume two time steps to clear A's way, reducing the payoff of A by -1 and making the final payoff [A=1, B=3]. The final payoff is calculated in the same way and is the average of the payoff recieved at the end of the two hierarchies. The final payoff in this case becomes [A=2, B=2] as shown in the table. In the third column, if we calculate payoffs of B and C. Since there is no competition, there is only 1 possibility for the two cars to cross. After the first time step when B is at (2,2) and C is at (2,1), B waits in the second time step for C to clear the way. B loses payoff by -1 and hence the final payoff becomes [B=2, C=3]. The other payoffs in the rest of the cases have also been calculated in the same manner and are compiled together in table 5.7.

5.2.1 Observation one: Car C is invariably a "parasite"

As it is visible, car A tries to encroach into a lane, which gives an opportunity to car C to take advantage of car A and get into the game. In table 5.3, payoff of A has been calculated when C is present and also when C is absent. Same has been done for C, we have calculated payoff of C when A is present as well as when A is absent.

The motive of calculating the payoffs of each of these design was to prove that while encroaching a lane, C or cars in the same lane as C act as parasites over A or cars behind A. Car C in general, in such situations, is always benefited by the presence of A. And since it helps A in no way, C acts as a parasite over A. Which seems very intuitive as well, because if C has a chance to gain benefit, as a player it definitely would prefer it. So there is a higher chance that any new car coming towards the intersection would prefer to take position of car C over car A.

When observing payoffs of C in the nine games, in about four out of nine cases, that is almost half of the cases, we find that payoff of C is higher with A present than when A is absent. That is, in cases 2 and 6 which are actually four cases overall as we have merged the cases, the payoff of C is higher when A is present than when A is absent. In four different cases, which are case1, 3 and both cases of 4, payoff of C is same whether A is present or not. This shows how C in almost half of the cases, takes help of A to encroach into the lane, increasing it's own payoff in the game. This is how car C is invariably a parasite over car A.

5.2.2 Observation two: Car A is an unwitting altruist

It is clear that C takes benefit from the presence of A while A does not seem to gain anything with the presence of C. When we observe the payoffs of A in the nine cases, it is found that in eight out of nine cases, payoff of A is lower when C is present than when C is absent and in just one case, it is exactly same whether C is present or not. In cases 1, 2, 4, 5 and 6, payoff of A is lower with C being present than when C is absent. While in case 3 payoff of A is same for the two situations. This shows A is neither being directly benefited with presence of C nor is aware of C's gain in payoffs due presence of A. This is why A is an unwitting altruist. It helps C in most of the cases and is not even aware of it. This is also shows why a person would prefer to be in car C's position over car A's position.

5.2.3 Observation three: Car A is indirectly benefited

The most interesting observation is when we look through the payoffs of car B. Even though it is expected that car B might not play much role in the dependency of A and C, a very different fact comes into the picture. When we notice the payoffs of car B, in eight out of nine cases, that is cases 1, 2, 4, 5, and 6, payoff of B is always lower when both A and C are present than when compared to just A being present. Which is an expected conclusion, since with C being present, the intersection becomes crowded and hence the decrease in payoff of B. Also the probability that B might even have to face competition with C increases and therefore the decrease in payoff of B is well justified with the presence of C.

But in the above Observation Two, we found that payoff of A is lower in cases when C is present than when C is absent. Here we notice that payoff of B is getting reduced with C's presence as well. Now we know from the start that B is always in competition with A in order to encroach. Since A and B are competitors, the reduction in payoff of B gives a slight benefit to A. Even though C's presence does not provide benefit to A in its payoff but since its presence reduces the payoff of A's competitor, A gets a little benefit with the presence of C.

Hence C may have been getting benefit from A, without A's notice, acting as parasite over A in encroaching the lane. We deduce here that C's presence reduces B's payoff in eight out of nine cases. If payoff of the competitor is reduced, then we know that chances of winning increases in a game. Since payoff of B reduces with C's presence, A is also getting an overall benefit in the game. Not in the case of payoff but still A gets a benefit from C's presence.

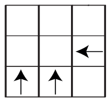
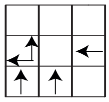
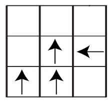
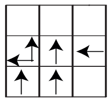
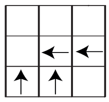
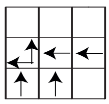
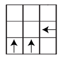
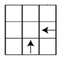
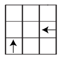
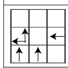
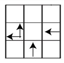
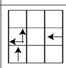
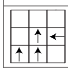
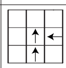
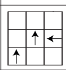
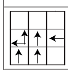
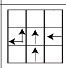
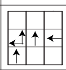
	Payoffs→ / Matrix↓	Car A	Car B	Car C
1.		1.5	1.5	3
2.		1.5	1.5	1.5
3.		1	1	3
4.		0.5	0.5	2
5.		-0.25	-0.25	1.75
6.		-1.25	-1.25	0.75

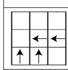
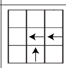
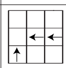
Figure 5.6: When payoffs of the three cars A, B and C in each the nine games is calculated using extensive form result in the above table. In this table, the nine games have been comprised into six cases. From previous calculation of the toy games, we know game two and game three deliver same payoffs for the three cars, hence their payoff are mentioned in one single row in the 2nd case of this table. Also games five and six and games eight and nine have same payoffs for the three cars and their payoffs have been mentioned in case 4 and case 6 respectively. The difference between the two games with same payoffs is the direction of the car at (2,2). The two arrows at (2,2) in each of the three cases 2, 4 and 6 correspond to this difference.

MATRIX 1			Payoff of Car A	Payoff of Car B	Payoff of Car C
			1.5	1.5	3
			2	2	-
			-	2	3

MATRIX 2			Payoff of Car A	Payoff of Car B	Payoff of Car C
			1.5	1.5	1.5
			2	2	-
			-	2	1

MATRIX 3			Payoff of Car A	Payoff of Car B	Payoff of Car C
			1	1	3
			1	1	-
			-	2	3

MATRIX 4			Payoff of Car A	Payoff of Car B	Payoff of Car C
			0.5	0.5	2
			1	1	-
			-	1	2

MATRIX 5			Payoff of Car A	Payoff of Car B	Payoff of Car C
			-0.25	-0.25	0.75
			1	1	-
			-	0.5	1.5

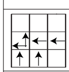
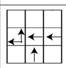
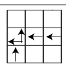
MATRIX 6			Payoff of Car A	Payoff of Car B	Payoff of Car C
			-1.25	-1.25	0.75
			0	0	-
			-	-0.5	0.5

Figure 5.7: Cars A and C are crossing the same lane and seem to be dependent on each other for the purpose. This game shows various possibilities whether the presence of car A affects the payoff of car C or not. Therefor, in each of the nine games, payoff of the car A has been calculated in both presence and absence of C. And payoff of car C has been calculated in both presence and absence of A. The payoff calculation for C suggests that payoff of C is always higher in the presence of A, which shows the parasitic behavior of C over A. Payoff calculation of A shows that payoff A is always lower in presence of C and hence shows altruistic behavior of A. Payoff of car B is always lower with the presence of C and since B is the rival of A, and as the payoff of the competitors goes down, A is benefited from the presence of C.

Chapter 6

Conclusion

We have been analyzing the question why cooperation occurs among a large group of people who even gain less out of cooperation. The particular situation taken by us is of a road intersection where there are no traffic lights to guide them. Different kind of people from such a large population cross the same street everyday, not knowing who they are crossing it with, and they still tend to cooperate. Every population has a group of defectors and cooperators, the cooperators pay a certain cost for the benefit of the other while defectors only gain from others loses. Without any explanation for the evolution of cooperation, the theory of natural selection favors defectors.

At a road intersection, if a person starts to defect assuming everybody else cooperates, one might end up with higher payoff. But since everybody is expected to think in the same manner, we would have a group of defectors creating a traffic jam. Even though the theory of survival of the fittest may favor defectors, we have a large population of cooperators around us. People cooperate even though their payoff is much lesser than what they would get by defecting. The cost that the cooperators pay here is measured in time because if people cooperate they will have to wait longer to cross. Since traffic jams are not that common, it proves that people tend to cooperate more than expected. Most of our games discussed above are based on competition, cars that do not defect in the game do not win and lose payoff over time.

With the help of the game theoretic analysis, we have been able to observe altruistic and parasitic behaviors of cars. Car A in our games is the unwitting altruist who helps car C in crossing the road. Car A gains no benefit in payoff by helping car C and still cooperates with it. Car C, on the other hand, acts as a parasite over car A. Car C takes advantage of car A's presence which is evident from the higher payoff of car C in presence of car A in almost all the cases. Even though the presence of car C may not help car A in payoff, C helps A in crossing the road by reducing the payoff of A's competitor B. We know by the end of the games that both the cars were dependent on each other for a better chance at crossing the lane.

While analyzing why cooperation occurs at places like road intersections, we come across the relation between cooperation and conflict. The cars seem to be gaining results not just out of conflict. Especially when we are trying to explain cooperation among large population, gaining result out of conflict seems more probable. In our road intersection, whether vehicles were crossing the road as a herd or alone, they all seem to gain result by blocking the main lane together with other vehicles. When people block the other lane by encroaching into it is mostly the solution to cross the road. Since vehicles from the main lane are aware that vehicles from the cross lane may defect or are going to defect is when people from the main lane move towards cooperation for better payoffs. The conflict between the two lanes leads to cooperation among the riders of the cross lane. Cooperation is a better way for the vehicles placed at positions of A and C to cross the lane.

Bibliography

[1] E. Pennisi, How did cooperative behavior evolve?, Science 309 (5731) (2005) 93–93.

[2] M. Nowak, R. Highfield, SuperCooperators: Altruism, evolution, and why we need each other to succeed, Simon and Schuster, 2011.

[3] A. C. Aktipis, Review of why we cooperate.

[4] L. Mui, Towards realistic models for evolution of cooperation (2012).

[5] S. F. Brosnan, F. B. De Waal, A proximate perspective on reciprocal altruism, Human Nature 13 (1) (2002) 129–152.

[6] C. Hauert, M. Doebeli, Spatial structure often inhibits the evolution of cooperation in the snowdrift game, Nature 428 (6983) (2004) 643–646.

[7] C. P. Doncaster, A. Jackson, R. A. Watson, Competitive environments sustain costly altruism with negligible assortment of interactions, Scientific reports 3.