Dissertation

Cesar Esparza

July 30, 2015

Abstract

The objective of this work is to describe in detail how a python library is built. The main purpose of this library will be to solve normal form games through an algorithm known as genetic algorithm (...) . In order to have a better understanding of how the library was built, it is considered important to give some theoretical background of game theory and evolutionary game theory.
Within this written work, it will be explained how evolutionary game theory works, application of the genetic algorithm, what considerations were made for building this specific genetic algorithm, etc.

1 Game Theory

Game theory or theory of games is a wildly known theory, which studies the interaction of decisions and has a very wide application in sciences like economics, political sciences and psychology. These topic of the French mathematician Emile Borel who in a note in his work La theorie du jeu et le equations integrals a noyau symetrique gauche (Borel 1921) mentioned:

The problems of probability and analysis suggest themselves concerning the art of war, or economic or financial speculations, are not without analogy with problems concerning games, though they generally have a higher degree of complication.

In 1924 Borel on his note On games that involve chance and the skill of the Players (Borel 1924) he mentions that the study of games that involve at once chance and the skill of the players appears to me similarly able to furnish an opportunity for mathematical research, the applications of which might far surpass the limits of the restricted domain to which this first study is limited. /such research might be extended to very many questions in which psychological unknowns figure along with algebraic unknowns. Then Borel continues saying that the only author who had studied problems with this focus was Joseph Bertrand in his Calcul des Probalites in 1889, distinguishing between mathematical and psychological aspects in an example given of a game of baccarat, but goes on stating why Bertrand study was incomplete (Borel 1924).

Game Theory was further researched and formally presented by the Hungarian mathematician John von Neumann in 1928 with his work Theory of Parlor Games, stating in the introduction that any event given external conditions and the participants situation (provided the latter are acting of their own free will) may be regarded as a game of strategy if one looks at the effect it has on the participants. (von Neumann 1928). And in 1944 John von Neumann and Oskar Morgestern published Theory of Games and Economic Behavior in which they established theory of games of strategy as an instrument to study problems in the economic behavior (Neumann, Morgestern 1944). Despite of the great contribution of von Neumann and Morgestern, game theory became widely used after the Doctoral dissertation from John F. Nash was published in 1950. The some of the main arguments from Nashs work are that he gives the possibility of analyzing games with more n-players, he introduces the concept of non-cooperative games having at least one equilibrium point and gives the idea of a dynamical approach to study cooperative games (Nash 1950). The last point I mentioned refers to how a cooperative game can be reduced to non-cooperative form, because the idea of a cooperative game stablished by von Neumann and Morgenstern was that cooperation was given under the assumption of in Nashs words players can communicate and form coalitions which will be enforced by and umpire.(Nash 1950) , but he proves this condition is not the only one that would define a cooperative game. The work from Nash opened the possibility of a broader application of the essential concept of game theory, many people have extended the fundamental concepts adopted from Nashs work and a lot of studies for the development of game theory have been made.

....

1.1 Normal form games

....

1.2 Nash Equilibrium

....

1.3 Agent-Based Modelling (ABM) and object oriented programming (OOP)

Generally speaking agents are created entities that represent entities of the real world, and make them interact to study their behavior. One of the first known scientists to show interest in the concept of entities was John von Neumann (Wilensky, Rand 2015). He thought in the future it would be useful to have artificial machines that could reproduce autonomously, in order to represent objects such as celestial objects (Wilensky, Rand 2015). Given to the suggestion of his colleague Stanislaw Ulam, von Neumann developed a simple model of cellular automaton. In which each cell take one of multiple states, and then the changes in it are based on the history of previous states from the cell and neighboring cells (Janssen, Ostrom 2006).

In 1970 Martin Gardner published a game by the british mathematician John Conway. Which was a simplified model of cellular automata applied into what he called Game of Life (Janssen, Ostrom 2006). In which an infinite universe is divided into cells, each cell interact with the neighboring cells (8 cells around it), according to a set of established rules creating complicated patterns according to simple states of the cells, which are alive or dead (Rendell 2001). In 1973 in a joint paper published in Nature (Nature Publishing Group) by Professor Maynard Smith and George R. Price describe the results of simulations made, which can be considered an example of an agent-based model (Maynard & Price 1973).

In 1971, 1978 the economist Thomas Schelling used a chessboard in which by moving pennies and dimes he represented what he described in his work Models of segregation (Janssen, Ostrom 2006). Finally another important work using agents were the simulations made by the political scientist Robert Axelrod in 1984 (Janssen, Ostrom 2006), in which he asked game theorists from different disciplines to submit a strategy which was used in the simulation of a tournament of one of the types of games called prisoners dilemma. Each strategy interacted with the other strategies, during a number of rounds (Axelrod 1984).

The previously mentioned works are some of the most relevant that involve this agent based modeling. Agent-based models (ABM) are being increasingly used to model complex systems.(Billari, et al 2006). Robert Axelrod in his book The complexity of Cooperation defines ABM as the simulation of agents and their interactions. This type of modeling is different from the traditional type, and it is a type of simulation that is viewed as bottom-up (Axelrod 1997, Billar 2006, Bonabeau 1994, Gilbert 1999) used for understanding properties of social systems (Axelrod 1997), by representing complex behaviours in the hopes of emulating some specific aspect. There are 3 ways for doing science according to Axelrod, which are induction, deduction and agent-based modeling, the latter starts like deduction with a set of explicit assumptions then it generates simulated data that can be analized inductively. And unlike induction the data that is produced comes from specific rules rather than real world data (Axelrod 1997). Joshua Epstein points ABM as a new tool for empirical research (Epstein, 2006). so it is very important to remember that our final goal with ABM is not to recreate reality, its rather for demonstrating a principal and understand how it works. There are at least three main components to take in account when building an ABM, the number of agents with characteristic variables (agents contain data together with methods which act of the data), a set of rules and the environment in which the interaction takes place (Brunn, 2007) these will be described in further detail in the description of the simulation.In agent based modeling what matters is not how decisions are made but how the interaction is given (Brunn , 2007).

A very important step when building a simulation is to choose a programming language. For the type of simulation that will be built, object oriented programming (OOP) is appropriate. In the recent years this language has gained popularity, even that it can be traced for a long way back. Before the concept of OOP was well established, all computing languages were procedural languages, which means that it looked like he program was all contained in one long procedure all data and the logic were presented in a long code.

OOP places attention on objects i.e. on properties, behavior or interaction with other objects (Pokkunuri) Object is the basic element. Objects possess attributes of procedures and data. Storing the data in variables and responds to messages by executing procedures (in Python called methods). The program is divided in individual objects (modules) each one can be viewed as an abstract data type. Each of the objects contain their own methods and data. (Pokkunuri). Communication between objects requesting action are usually called message. The purpose of this is that each object represent parts of whole program, but by breaking it down in modules each can be thought of as a particular action which we can relate in an easier how ideas and actions are structured in our everyday life. As an example we can think of an apple which would be our object and this apple contains attributes such as color, size, weight, etc. but it can also contain methods such as growing, changing color, falling from a tree, etc. This way of conceptualizing ideas in such conventional way in relation to our everyday life make it a perfect candidate for using in simulations. In this project OOP would simplify structuring the ABM, in which it was mentioned before we have so very defined concepts (agents, rules and environments).

To be able to work with OOP there is one idea we need to understand properly, which is the difference between class and object. A class can be thought of as a general description or blueprints of something but is not the thing itself. And what the class is defining are abstract ideas of what was mentioned before, attributes and methods, formally a class is defined as a template from which objects are created (Dyke, 1989). The next concept is object, would be the "tangible" instance of the initial template which is the class(Luna 2012). To understand this correctly we can think of the class as the blueprints of something we want to build, and the object is that thing we wanted to build from the blueprints. And when creating an object the abstract ideas from the class, become specific characteristics of the object. A relevant property is inheritance, which allows a class to inherit methods and attributes from another class, this makes the class that inherits a subclass of the class it inherits from, its important to mention that this subclass can redefine inherited methods and can add methods that can differentiate it from the class it inherited from (Dyke, 1989). Some other relevant properties from OOP are dynamic binding, which is not exclusive for OOP, means that the binding of operator to a particular operation takes place at the run time. Encapsulation is another and it describes the scope of unrestricted reference to the attributes of an object. An object can examine and modify its own attributes, and allows access to its attributes to other objects through accessing functions allowing it to have control over any changes requested from other variables. Data abstraction refers to how any object can be required for any information, and the fact that who requests gets what it asked for. (Pokkunuri, Dyke 1989)

....

1.4 Evolutionary game theory

Evolutionary game theory has its roots in evolutionary biology. Even if this overview is not intended to be focused in biology some remarks from it are worth mentioning. From the 6th edition of Origin of species Charles Darwin outlines that in nature there exist many struggles for existence. Some examples are the struggles of a species in the nature, between species and within species. Darwin stresses that the most severe struggle might be within species if they become into competition (Darwin 1872). Darwin thought of life as a game (Vincent et al 2005). In origin of species with the idea of all possible interactions of the organic beings mentions that there may be infinite varied diversities of structure for each being under the changing conditions of life, and continues saying that through the course of generations, there can occur variations that can give perhaps a slightly advantage to some beings giving them a higher chance of survival and procreation. The preservation of favourable characteristics and destruction of what he called injurious, he called it natural selection or survival of the fittest (Darwin 1872). I go this far because later on we can think of the concept of fitness, as the utility a player has in evolutionary game theory which in general terms can be called the fittest.

The ideas of Darwin were presented as logical verbal arguments, today evolution is viewed in terms of genetics. The first edition of Origin of species was in 1959 without the understanding of genes (Vincent et al 2005).

Fisher might have been the first to apply game theory to evolution in his study on sex ratios in 1930 (Pallen 2009) although at this time the formal definition of game theory had not been presented yet. Later in 1961 Lewontin discussed species playing against nature, and said that species should adopt the maximin strategy if nature presented worse-case scenarios (Maynard 1974, Sigmund 2004). In 1967 in an the article Extraordinary sex ratios wrote in Science William D. Hamilton uses game theory to model local competition and frequency-dependent fitness values (Sigmund 2004). Perhaps the most important contribution in evolutionary game theory was the one made by Professor John Maynard and Dr. George Price. John Maynards attention was first caught by an article written by George Price in 1968 for Nature about ritualized behavior in animal contests (Maynard 1976) unpublished for being too long. Then in 1973 John Maynard and George Price published a joint paper On the logic of animal conflict in which the mathematical concept of an evolutionary stable strategy is established, and it applies concepts of game theory to the study of conflicts between animals (Sigmund 2004). The idea Maynard and Price presented was that concepts from game theory could be used to characterize eventually stable endpoints in the evolutionary process, with the concept of evolutionary stable strategy (McNamara 2010). It can be said that the concept of evolutionary game theory was born with the ideas of Maynard and Price.

Nowadays evolution by natural selection can be thought of as a game, where some behavioural patterns (often referred to as phenotypes the equivalent to strategies when related to traditional game theory) from animals are more successful than others (Carmichael 2005). Now I can relate evolutionary game theory with traditional game theory in the some essential concepts. Animals in the biological concept are equivalent to the players (agents) that participate in a game; the environment in which animals interact is comparable to the set of rules that regulate interaction in the traditional form; as mentioned before the heritable phenotypes of animals can be thought of as the strategies that players use in the traditional form; a tricky concept to relate to evolutionary game theory is the one of payoffs (utility) in traditional game theory for this I will refer to how Maynard and Price define it in The logic of animal conflict as the contribution the contest has made to the reproductive success of the agent (Maynard & Price 1973) which could be the expressed in terms of fitness (Darwin 1872) the fitness in an agent directly influences the frequency of the strategy in the population (Vincent 2005), Maynard and Price take in account three factors to be taken in account: the advantages of winning compared to losing, disadvantage of being seriously injured and disadvantage of wasting time and energy in the contest (Maynard & Price 1973) this are not usually considered in games but I consider important mentioning. Another very important concept is equilibrium, in some evolutionary games the existence of evolutionary stable strategies (ESS) , I will not yet talk about the mathematical implications of ESS. Roughly we can consider an ESS as a strategy that predominates in frequency in evolutionary games and that in the case of the emergence of a mutated strategy is not invaded (threatened to be reduce in number) this concept of ESS has similarities with the concept of Nash equilibria as seen by in the sense that both can be no-regret strategies when in a population a Nash or an ESS is played no individual can benefit from unilaterally changing their strategy (Vincent 2005). According to T.L. Vincent, resistance to invasion is only one of the two notions of an ESS, the second notion is convergence stability which implies that a population evolves to an ESS when its strategy composition is near but not at the ESS mentioned by Eshel in 1983, the implications of this will be discussed later.

Thinking of how the idea of evolutionary game theory was conceived from behavior of animals, comparing it to the traditional game theory, there are some characteristics that distinguish them from each other. First and perhaps one of the most relevant assumptions in traditional game theory is that every player is rational which means they make rational decisions to maximize their profits, and also they are aware of the possible payoffs of the other players and that other players are rational, also the rational players are aware of the game rules, evolutionary game theory does not make such assumption of rational players, instead the strategies are hard wired to them. Traditional game theory it is about choosing from different strategies looking to optimize the payoffs, whilst evolutionary game theory is to determine strategies that will endure through time. Traditional game theory as said before has a set of strategies from which it can choose, whilst in evolutionary game theory the strategies are already defined given that they are inherited although there can be present some occasional mutations. Also evolutionary game theory there will be groups of players that possess the same set of strategies and the same related payoff from these strategies, in traditional game theory each player has their own set of strategies and their own associated payoffs per strategy (Vincent 2005). The application of evolutionary game theory in different areas of study has grown, and with this some assumptions change. For example in the biological application players do not choose their strategies and never change them, unlike in the economic application the players are people, who can choose and change their strategies (Samuelson 1997).

What is evolutionary game theory? I would define it as the combination of some game theoretical concepts, with the concepts of natural selection in evolution. It is a change in the focus from traditional game theory, because the main goal of evolutionary game theory is to observe the stable equilibria and how it changes through time with the different interactions between strategies, instead of only focusing in optimizing outcomes for a single game. In this sense we can identify two main approaches to evolutionary game theory (McKenzie

2009). The first approach is the static approach which is directly derived from the work of Maynard and Price, the main tool for analyzing is the ESS. The second approach through the study of the population dynamics (change in density of existing strategies) and of how the strategies evolve in the model built.

1.5 Evolutionary Stable Strategy (ESS)

Symmetric games

1.5.1 How we use it

Evolution can be thought as a game. According to the *theory of evolution* the existing organisms are the result of many different selections, in which the ancestors resulted the most fitted. These selections were the result of the interaction between the organisms of the same and other species. I dare to say that most of the times it reduces to hunters and preys. Where the best hunter survives and the best prey survives. But what does this imply? how can one be a good hunter? even more how can one be a good prey?. Different organisms posses different attributes, which determine to what part of the chain they belong. But competition for survival is not only seen between different organisms, it is also seen within the same species. Within same species there are important considerations one of the most important being the capability to reproduce.

For the purpose of this work, we will be focusing on the interaction between the same species. The model we will use is simplified.

2 Python and genetic algorithm(relating evolutionary game theory)

....

3 Library (Package)

The library is the file that contains all the components of the code. Because of the object oriented properties that Python facilitates, the code is segmented into 4 modules. According to their function the order in describing each is not relevant.

The population module contains the instructions to create a class named Agent (why agent described in ABM)the class has an initialization(__init__) method that takes as parameters strategies, utility and the possibility to add a label. After the initialization, another method is presented increment_utility which is set for incrementing the agents utility, the criteria for this increment_utility will be explained in another module.

- **Strategies:** Each created agent will be assigned a strategy

- **Utility:** The utility each created agent generates after each interaction with another agent.
- **Label:** The possibility of adding a label to each created agent, to track their performance.

The environment module for this project the representation of the environment has only two characteristics. It is set to make two agents interact pairing them randomly and it also sets the rules which these paired agents use to interact. Some of the methods contained in this module make use of a module named random from python. This module is imported when the environment module is executed.

Environment module creates a class BiMatrixRandomEnv, named after the characteristics bimatrix and random environment. This class has an initialization (__init__) method that takes as parameters number_of_agents and bimatrix, within the initialization the some variables are defined, these variables along with the parameters will now be explained:

- **Number_of_agents:** Input by user, total population of agents regardless of the type of agent(i.e. row agent or column agent).
- **Bimatrix** Input by user, bimatrix of payoffs(can be symmetric or assymetric).
- **Number_of_row_agents:** Result from dividing by 2 the previously input value number_of_agents. And gives the number of row agents.
- **Number_of_col_agents:** Result from dividing by 2 the previously input value number_of_agents. And gives the number of column agents.
- **Number_of_row_strategies:** Number of strategies that will be available for row agents. Calculated by counting how many rows the bimatrix has.
- **Number_of_col_strategies:** Number of strategies that will be available for column agents. Calculated by counting how many columns the bimatrix has.
- **Row_strategies:** List containing the available strategies for row agents.
- **Col_strategies:** List containing the available strategies for column agents.
- **Row_agents:** Instances of class Agent are created according to the number_of_row_agents.
- **Col_agents:** Instances of class Agent are created according to the number_of_col_agents.

After the initialization, a method interact is defined. This method first defines a variable called pairs which is assigned to a function randomly_pair_agents that will be explained later. It also contains a for loop this loop within other things contains a variable which is set to a function strategies_to_utilities which will be explained, variables in the for loop are the following:

- **Ra**
- **Ca**
- *textbfPairs*
- *textbfUtility*
- *textbfAgent.increment_utility*: The increment utility function is defined in the population model. The structure for in this for loop is as follows: agent.increment_utility(utility[x]) and what it does is assign the function increment utility to an agent can be ra (row_agent) or ca (col_agent), the utility in parenthesis was assigned in the previous variable, and it is only calling the value with the position x in the list. Given that we only have 2 types of players (we are using a bimatrix) x can be either 0 or 1.

Following interact the method previously mentioned strategies_to_utilities is defined. This method is in charge of obtaining the specific pair of utilities (assigned to row and column) from the bimatrix. It then returns these values to the utility variable in the interact method and interact uses it to assign the utilities to each agent.

After strategies_to_utilities the method randomly_pair_agents is defined. This method is used by interact too, and what it does is that the previously created row and column agents that are contained in lists are randomly selected (one of each type) and then paired so they can interact.