Studium licencjackie

Kierunek: Metody Ilociowe w Ekonomii i Systemy Informacyjne

Forma studiów: Stacjonarne

Imie i nazwisko autora: Kacper Mordarski

Nr albumu: 101247

<tytu>

Praca licencjacka napisana
w Katedrze Matematyki i Ekonomii Matematycznej
pod kierunkiem naukowym
dr hab. Michaa Ramszy

Warszawa 2022

Contents

1 Introduction

1.1 General Introduction

This paper aims to replicate, to some degree, the seminal paper of **?** on the emergence of cooperation. The secondary goal of this research is to provide the publicly available code allowing replication of the actual results.

» To co jest poniej to do poprawki. Trzeba tutaj napisa taki standardowy wstp machany rkami. Koniecznie trzeba si pozby “The paper”, to jednak jest nie do przyjcia.

either prove or disprove the research paper with title **"Scale-Free Networks Provide a Unifying Framework for the Emergence of Cooperation"** written by F.C. Santos and J. M. Pacheco and published in Physical Review Letters nr 95 (later on reffered to as "The Paper"). Regardless of the outcome this work is going to be useful because the above mentioned authors did not provide the code that was used to conduct the necessary simulations. Therefore goal is not only to disprove the above mantioned paper but also to provide readers with clear and easy to follow code.

1.2 Description of The Paper

The Paper was published in “Physical Review Letters” on the 26th of August 2005 and, according to Google Scholar, has been since cited over 1600 times. It clearly shows the magnitude of the said paper and its groundbreaking character. This paper presents the results of simulations conducted by the authors and their implications for evolutionary game theory.

The crucial thing to understand is that the authors of The Paper changed the approach to modeling such games by applying Scale-Free Networks of Contacts (later on referred to as "SF NOCs"). Its innovativeness lies in the never used before degree distribution of said graph. Before being used graphs had a degree distribution with a single peak. It means that every "player" could interact only with a fixed number of other "players". SF NOCs are said to perform better at modeling the actual, existing societies and networks. It complies with the rules of growth and preferential attachment (rich gets richer). When we analyze some actual networks, for example, the Twitter network, we observe that those with a bigger count of "followers" are more likely to gain new ones than accounts with a low count of followers.

One of the goals of The Paper was to compare the results of simulations on different kinds of graphs. According to The Paper, players that occupy vertices of SF NOC are much more likely to cooperate than on any other graph. Those results came up both in the Snowdrift game (later on referred to as SG) and Prisoners Dilemma game (later on referred to as PD).

» Tutaj opisa jaka jest struktura caej pracy, w sensie w tym rozdziale to jest to, w tamtym rozdziale to jest tamto. Robimy to na kocu.

2 Elements of the noncooperative game theory

» Tutaj trzeba opisa wszystkie koncepcje matematyczne zwizane z teori gier. Wydaje si, e to co jest nam potrzebne to definicja gry w postaci normalnej i definicja równowagi Nasha. Na koniec zamieci przykady pojedynczych gier (ale niekoniecznie PD i SD. Poza tym warto chyba jednak opisa jaki ogólny framework, w którym gra jest populacyjna, co w takiej sytuacji jest mierzone i jak to ewoluuje w czasie. Myl, e tutaj mog Panu pomóc. Dodaem tutaj dwa cytowania typowe dla modelu: **???**. Typowe cytowanie do teorii gier to **??**.

The noncooperative game theory is the basis for this dissertation. For a game to be noncooperative means that all of the game participants (players) act in their own best interest and therefore compete with each other. Moreover, they need to be economically rational — choose a strategy based on its optimality. Games in this paper will be presented as normal-form games. For the normal-form game to be described, it requires a couple of elements. First of all, let us define a set of players $i \in I$, where I is a finite set of natural numbers (i. e. $I = \{1, 2, ..., N\}$). For each player i, we define the pure strategy space S_i consisting of k pure strategies. The sequence of strategies — $(s_1, ..., s_k) \in \prod_{i \in N} S_i$ is called the strategy profile of a player. Lastly we define function (denoted as $u_i(s)$). This function maps the von Neumann-Morgenstern utility (payoff) for each profile of strategies for a given player. Mathematically speaking we can denote this function as $u_i : \prod_{i \in N} S_i \to \mathbb{R}$. The last two assumptions of a normal-form game are that the structure of the game is perfectly known — each player knows all of the possible strategies and utility functions in the given round of a game and that players have no knowledge of other player choices. It means that each time they select their strategy simultaneously, thus cannot respond to the actions of other players.

The next important part of the game theory is known as the Nash equilibrium, introduced by Antoine Cournot and furtherly developed by John Nash in his famous article **?**. The general idea behind this theory is that in some games there exist an equilibrium in which none of the players has an incentive to alter their strategy. By a lack of incentive we reffer to a situation in which one player, given set of strategies chosen by the other players, cannot achieve higher payoff (utility) by altering his strategy.

In a mathematical framework we define the Nash equilibrium as follows:

Let S_i be be the set of all possible strategies for player i, $s^* = (s_i^*, s_{-i}^*)$ be a strategy profile (set containing one strategy for each player — s_{-1}^* denotes strategies of all the other players except of i). To represent the incentive of players we use utility function $u_i(s_i, s_{-i}^*)$. It is a set of all possible payoffs for player i with fixed strategies used by all of the other players. For the strategy profile s^* to be the Nash equilibrium, it must satisfy the inequality:

$$\forall s_i \in S_i : \; u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*)$$

This version of Nash equilibrium is called weak. It means that the player i is indifferent to two or more strategies. For the Nash equilibrium to be strong, it must satisfy the strict inequality:

$$\forall s_i \in S_i : \; u_i(s_i^*, s_{-i}^*) > u_i(s_i, s_{-i}^*)$$

It is important to note that obtaining an equilibrium in a Nash sense does not guarantee the optimality of the solution. To be specific, it means that payoffs may not be Pareto optimal **?** thus, there may exist strategy profile with a larger payoffs.

Analysis of example games.

"Battle of sexes"

The game introduced by R. Duncan Luce and Howard Raiffa in their book **?** is a classical decision analysis problem. Let us consider two music critics. They agreed on attending one of the two excellent concerts heppening in their whereabouts. They can either go to a Shostakovich or Stravinsky concert. One critic — let us call him player 1 would prefer to go to the Shostakovich concert. Meanwhile, the other critic (player 2) would rather attend the Stravinsky concert. Unfortunately, they have no way of contacting each other, and therefore, the decision of where to go must be undertaken simultaneously. We have two players, each with two possible strategies - Shostakovich, Stravinsky. Each of them achieves a payoff of 1 if they get to go to their place of choice. Furthermore, if they both choose to go to the same event, they gain utility from spending time together. We can represent this game in form of a payoff matrix as follows:

$$\begin{array}{c|c|c|}
 & Shostakovich & Stravinsky \\ \hline
Shostakovich & 3,2 & 1,1 \\ \hline
Stravinsky & 0,0 & 2,3 \\ \hline
\end{array}$$

In this matrix the player one chooses the row strategy, and the player number two chooses the column strategy. Each cell represents two payoffs — letf one is first players' payoff, the right one represents second players' payoff. There are two Nash equilibriums in this game. They occur for the following strategy profiles: Shostakovich, Shostakovich, Stravinsky, Stravinsky

"Stag Hunt"

The stag hunt game originates in a work of Jean-Jacques Rousseau **?**. It represents the conflict between social cooperation and conflict. The idea behind this game is that there are two hunters who can either hunt for a stag or a hare. Moreover, none of the hunters can singlehandedly take down a stag — it is a job for two. If either hunter chooses to go for a stag and the other chosses a hare, they get a payoffs equal to 0 and 4, respectively. If they both go for hare, each one of them gets a payoff of 2 (there are only so many hares that they can provide an accumulated utility of 4). However, if they both decide to go for a stag, they get a payoff equal to 5 respectively. Therefore the payoff matrix looks as follows:

$$\begin{array}{c|c|c|}
 & Stag & Hare \\ \hline
Stag & 5,5 & 0,4 \\ \hline
Hare & 4,0 & 2,2 \\ \hline
\end{array}$$

As we can see, this game also has two Nash equilibriums — strategy profiles {Stag, Stag} and {Hare, Hare}. Clearly the equilibrium {Stag, Stag} is Pareto optimal.

?

We are going to be discussing noncooperative games played by two players with pure strategies.

3 Elements of the graph theory

» Tutaj musimy opisa wybrane elementy teorii grafów. Myl, e pojcie grafu nieskierowanego oraz wybrane grafy losowe, razem z cytowaniami. Tutaj mona zaczy przykady zarówno grafów losowych ale równie maych grafów losowych tak aby byo jasne jaka jest struktura tych grafów. Doda oczywicie odpowiednie cytowania. 40,5

The next important element of our work is the graph theory. We can define a graph as two sets — one containing objects called vertices and the other containing pairs of vertices called edges. **?**. A single vertex is usually denoted by a natural number. It can stand for various objects, but in this dissertation, we'll use the vertices of a graph as a representation of players. The edges epitomize relations between vertices. In our case, the vertices connected via an edge stand for players connected, thus able to engage in a game. We can distinguish between two types of a graph — directed and undirected ones.

A directed graph not only shows relations between vertices but also specifies the direction in which the relation occurs. As an example of a directed graph, we can give a simple illustration of an electric circuit **?** [PRZYKADOWY OBWÓD]. In those graphs, the notation of edges is crucial since the edge $\{2, 3\}$ is not equal to the edge $\{3, 2\}$ (the relation is in the opposite direction).

For undirected graphs, this is not the case. Since edges represent relations with no regard of their direction, we can denote the relation between vertices a and b either as $\{a, b\}$ or $\{b, a\}$. As an example of an undirected graph, we can think of a graph in which vertices represent humans and the edges represent the relation of the fact that they live in the same city. [WSTAWI GRAF, 5 wzów] Since this relationship works both ways (if subject a lives in the same city as subject b it implies that subject b lives in the same city as subject a), we do not need to denote the relation as $\{a, b\}$ and $\{b, a\}$, but we can use more convenient denotation and just use one of those. Since we do not use directed graphs in this dissertation, henceforth, as a graph, we refer to an undirected graph.

Let us define a neighborhood of a vertex in a graph. The neighborhood of vertex v in a graph G is a subgraph of G, consisting of all vertices adjacent to the vertex v and all of the relations (edges) between them. Two vertices are adjacent to each other when they are connected via an edge. By neighbors of a vertex v, we denote all of the vertices adjacent to the vertex v however, we do not care for relations between them. This concept is crucial for our work because it will allow us to identify all of the vertices able to engage in a game with a previously chosen player. The degree of a vertex n is a number of all vertices connected to the n by an edge. The important, in our case, characteristic of a graph is its' degree distribution. The degree distribution is the probability distribution of degrees in a graph. This characteristic varies throughout the different types of graphs. For a regular graph, it is a single value with the probability of 1 (because in a regular graph all of the vertices have the same connectivity).

In our work we will be using a type of graph called Scale-Free Network. To generate this type of graph we will be using Barabasi-Albert Model **?**.

In order to describe the procedure of creating such a graph we must mention a preferential attachment and growth rules. The growth rule means that, to a given extent, the graph evolves (grows) with time. The preferenctial attachment rule applies to adding new vertices to an existing graph. It defines the probability of connecting the new vertex n to the preexisting vertex i. The probability is given as follows:

$$p_i = \frac{k_i}{\sum_j (k_j)}$$

where k_i is a degree of vertex i and $\sum_j (k_j)$ is a sum of degrees of all existing nodes (which, since our graph is an undirected graph, is equal to twice the amount of existing edges). This rule clearly favouritises the edges with higher connectivity and causes our graph to have so called hubs. A hub is term for a vertex with considerably higher degree then the other vertices. Thanks to this rule the degree distribution of a graph is given by the formula:

$$P(k) \backsim k^{-3}$$

which makes it a scale-free graph.

The procedure of building a Scale-Free network using Barabasi-Albert model is as follows **?**:

```
Data: parameters of Barabasi-Alber model — avarage connectivity, number of
      initial nodes, number of final nodes
Result: Scale-Free network with a given number of vertices and avarage
        connectivity
initialization;
while number of vertices less than the set number do
    add new vertex in compliance with preferential attachment rule;
end
```

Algorithm 1: Barabasi-Albert Model

4 Cooperation among the players

» W tym miejscu w kadym rozdziale trzeba wpisa krótkie streszczenie co jest w danym rozdziale.

4.1 A brief description of Snowdrift and Prisoners Dilemma games

Prisoners Dilemma Game. The Prisoners Dilemma game is a widely known problem in game theory and decision analysis. It shows situations in which the outcome is not optimal, even though players act in their own best interest. In the scope of our analysis, it is essential to note that in the PD game, the best strategy is to defect, regardless of the opponent's choice. The game is parameterized as follows:

$$\begin{array}{c|c|c|} & C & D \\ \hline C & R,R & S,T \\ \hline D & T,S & P,P \\ \hline \end{array}$$

where the values are given as follows:

$$\begin{aligned} &T = b > 1, \quad R = 1, \\ &P = 0, \qquad\quad S = 0, \\ &1 < b \leq 2. \end{aligned}$$

We see that the above restrictions order the parameters as follows:

$$T > R > P = S.$$

Snowdrift Game. The Snowdrift game represents a metaphor for cooperative interactions between players. Contrary to the PD game, the Snowdrift game stimulates cooperative behavior amongst players. It doesn't make the deflection strategy inapplicable, but the game's payoffs encourage cooperative behavior more than the payoffs of the PD game. The optimal strategy is to cooperate when the other defects and to defect when the other cooperates. The game is parametrize as follows:

$$\begin{array}{c|c|c|} & C & D \\ \hline C & R,R & T,S \\ \hline D & S,T & P,P \\ \hline \end{array}$$

where the parameters' values are:

$$T = \beta > 1, \quad R = \beta - \frac{1}{2},$$
$$S = 1 - \beta, \quad P = 0.$$

We see that the above restrictions order the parameters as follows:

$$T > R > P > S.$$

Nash equilibria. » Tutaj analiza równowag Nasha w opisanych powyej grach w one-shot games. Pokaza, e jest albo nie mam kooperacji. Tutaj równie wprowadzamy gr na grafach losowych, a wic musimy opisa na jakich grafach, itd. Tutaj równie opis uczenia si ale w sensie algorytmu matematycznego nie implementacji.

4.2 Replicator dynamics

1. The site x is updated.
2. A neighbor y is drawn at random among all k$_x$ neighbors
3. if cumulative payoffs of y (P$_y$) are greater than cumulative payoffs of x (P$_x$), the chosen neighbor takes over site x with probability (P$_i$) given below.

5 Algorithms and simulations

This chapter contains a description of the algorithm used in the simulation.

» To co powyej do rozszerzenia. Generalnie ten rozdzia ma zawiera implementacj koncepcji matematycznej z poprzedniego rozdziau.

In this section of the dissertation, I'm going to describe the key elements of the algorithm used to replicate the results obtained in The Paper. My understanding of the mechanisms on which the algorithms are based is limited to the rather vague and unclear description in The Paper.

5.1 Functions

In order to perform necessary calculations I had to define the following functions:

1. `Transform` — This function is used to map strategies onto a vector of arrays. As an input this function takes one of the edges of a SF NOC, as an output it returns a vector of length two (two vertices connected with an edge).

2. `Strat` — This function is used to map previously distributed strategies onto a vector of edges. As an input it takes an edge and as an output it returnes a vector of length two (two strategies previously attributed to the vertices).

3. `Games` — This function is used to evaluate the results of games played between players (vertices connected with an edge). As an input this function takes a vector of edges, vector of strategies and a vector of accumulated payoffs. It returns an adjusted vector of accumulated payoffs.

4. `CheckStrat` — This function fulfills a number of tasks. It takes as an input a randomly chosen vertex, vector of strategies, vector of accumulated payoffs, and the SF NOC. It identifies all neighbors of the previously mentioned vertex, then shuffles them, and then looks for a neighbor with a different strategy. Then it proceeds to change the strategy of the vertex with lower accumulated payoffs. The probability of the transition is described in section 1.5.

5.2 Description of the algorithm

The main algorithm is the fundamental element of this dissertation. It consists of 3 nested loops executing instructions necessary to conduct simulations, on which my research is based. I will be describing those loops in an inside-out order.

The first loop is responsible for evaluating the results of the games, potentially changing the strategies of players, and keeping track of the proportion of the strategies used by players. To run properly, it requires previously set parametrization and a set of edges of the Barabasi-Albert graph. As a result, it produces a vector of length 100 in which elements are proportions of coop/def strategies, measured after each generation (1001–1100) in the population of players. This loop is repeated 2 000 times, which gives us a total of 2 200 000 generations for one Barabasi-Albert graph parametrization for a given game. The loop itself operates as follows:

```
Data: Barabasi-Albert Moedel, array of edges, array of strategies, array of
      accumulated payoffs (initially equal to 0), empty vector of length 100,
      parametrization of the games
Result: Vector of length 100 containing coop/def proportion for the population
initialization;
for k in generations do
    Play a single round of a given game between all possible pairs of players by
      applying function games on a vector of edges.;
    Randomly choose one vertex. ;
    Attempt to change the strategy of the previously chosen player (or one of his
      neighbors) by applying function CheckStrat. ;
    if k is greater than the number of generations minus 100 then
        Evaluate the coop/def proportion in the entire population;
        Save the result to the appropriate value of a vector.;
    else
    end
end
```

Algorithm 2: The inner loop

The outer loop to the one described above is responsible for “resetting” the Barabasi-Albert graph. Since separate simulations are supposed to be carried out on a randomly generated SF NOC (but with the same parametrization), we need to conduct 100 of them (for each payoff parametrization); this loop is an indispensable element of the algorithm. The crucial fact to note is that this loop is also responsible for creating an object named `ArrPath` — vector of length 100, used to store the results of the simulations. The procedure followed by this loop is as shown below:

```
Data: Barabasi-Albert parametrization
Result: Scale-Free Network built by Barabasi-Albert Model, list of edges, list of
        strategies, vector of accumulated payoffs, ArrPath — an array of length
        100 storing results of the simulations
initialization;
for j in simulations do
    Create a Scale-Free Network by using the Barabasi-Albert model with
      previously set parameters.;
    Create an array of length equal to the number of players, containing their
      strategies. ;
    Extract an array of edges from a previously created graph by mapping
      function transform onto a list of edges. ;
    Produce an array of tuples representing strategies for every player by
      mapping a function strat onto an array of edges. ;
    Create an array of accumulated payoffs (initially filled with zeros). ;
    Generate a vector of length 100 filled with zeros, called track. This vector
      will be later on used to store coop/def proportions. ;
    if j equals to 1 then
        Create an object called ArrPath with one element equal to track;
    else
        Merge the current track with ArrPath. ;
    end
    Initiate the inner loop described above. ;
end
```

Algorithm 3: Second loop

```
Data: None
Result: Parametrization for the games, DataToSave — an array of vectors that
        will be used to store and ultimately save our results.
initialization;
for i in number of parametrizations do
    if i is equal to 1 then
        Create an array of length 20 — DataToSave. ;
    else
        Store the current ArrPath as a value of DataToSave. ;
    end
    Set up payoffs of the games based on the current iteration. In each iteration,
      add 0.05 to the current values of the payoffs. ;
    Create a matrix of payoffs for the games with current parametrization. ;
    Initiate the second loop. ;
    Save DataToSave to the local repository. ;
end
```

Algorithm 4: Third loop

5.3 Simulations description

Because of the nature of this paper (an attempt to clone the results of The Paper), our simulations must be conducted in strict accordance with the methods outlined in The Paper. Therefore it is only natural that we must follow each step with the utmost care and diligence. According to The Paper, we must conduct 100 simulations for each parametrization. Each simulation is performed following those steps: Where the parametrization is as follows:

1. Setting up the parameters which are needed to create SF NOCs (such as the number of final vertices (population size), the average connectivity, etc.).

2. Choosing the parameters of the game which is to be simulated (either PD or SG).

3. Creating the randomly generated SF NOC (we use Barabasi - Albert model to do that). The SF NOC must be created in compliance with preferential attachment and growth rules.

4. Randomly distributing strategies amongst the population (SF NOC in this particular case). Each vertex can either get a cooperation or deflection strategy.

5. Each pair of cooperator-deflectors engages in a round of a given game. In compliance with replicators dynamics, we keep track of cumulative payoffs for both strategies so that "players" can adjust their strategies throughout the population. This step is repeated 11 000 times, each time is called "generation". The first 10 000 is the so-called "transient time".

6. We collect results (equilibrium frequencies of cooperators and defectors) by averaging over the last 1 000 generations.

» Poniej wrzuciem przykadowy algorytm w pseudokodzie. Myl, e to jest to co chcemy wykorzysta aby zapisa sam algorytm. To oczywicie musi by potem opisane w tekcie.

5.4 Compiling LaTeXfiles

```
Data: this text
Result: how to write algorithm with LaTeX2e
initialization;
while not at end of this document do
    read current;
    if understand then
        go to next section;
        current section becomes this one;
    else
        go back to the beginning of current section;
    end
end
```

Algorithm 5: How to write algorithms

6 Results and discussion

» Tutaj wpisujemy uzyskane wyniki oraz dyskutujemy je, np. porównujemy do wyników z oryginalnego artykuu.

7 Conclussions

» Tutaj na samym kocu dopisujemy dlaczego i co zrobilimy oraz jakie mamy dalsze plany na badania.

List of Tables

List of Figures

Streszczenie

Tutaj zamieszczaj Pastwo streszczenie pracy. Streszczenie powinno by dugoci okoo pó strony.