National Research University Higher School of Economics

Faculty of Economic Sciences

Bachelor’s programme 'Economics'

Bachelor Thesis

Numerical Approach to Locating Multiple Equilibria of Game-theoretic Models

by Dmitrii Rykunov

student of group BEC139

Advisor: Ivan Stankevich

Master, Lecturer

Moscow 2017

1 Contents

2 List of Algorithms

3 List of Figures

4 Abstract

This thesis investigates possibility to develop a general-purpose framework to numerically analyze models involving decision-making aspect in terms of system stability and individual decision-makers' behavior. It focuses on models equivalent to finite non-cooperative games. We start with a mathematical formalization of such models and their equilibria and outline of an approach to analyze them, based on evolutionary game theory concepts. We proceed by implementation of the approach in an algorithm, inspired by evolutionary optimization algorithms, capable of locating multiple equilibria with iterated local search procedure. The algorithm's convergence to equilibria concepts traditional for evolutionary game theory studies, namely evolutionary stable strategy and evolutionary stable set, is demonstrated both by theoretical findings and by empirical results.

5 Introduction

For the whole history, humanity was dependent on its ability to cooperate and in general, its progress tightly bounded to a great variety of social interactions. Therefore, to the known extent people always wondered what people would do in a given context or circumstances.

We could say that to survive it was necessary for human species to predict outcomes of social interactions. From the very ancient times – prehistoric tribes, collective hunting, great wars, conquests and slavery – to modern times – literacy spreading, science developments, overall quality of life improvements – all those activities demanded and continues to demand great level of cooperation and coordination between humans. To achieve that people need to negotiate with each other and consider carefully reciprocal strategic opportunities. Results of it could be found on any scale of social structure. Psychologists made a great work on thorough depiction of the dynamic of a child adopting these basic social skills that in everyday life left almost unnoticed, because it is so usual, that we take it for granted. We could notice their application on the smallest scale around us – we usually unconsciously relate possible actions of people around us to the most primitive social constructs as social norms in everyday interactions; and on a greater scale we relate to legal system, law and contracts in more complex, long-term interactions as on the level of state and between states. Nowadays globalization process pulls us toward the level of worldwide interaction.

The truth is that the greater the complexity of social interaction being considered, the harder it is to maintain the desirable level of clarity of matter under consideration, hence desirable accuracy of prediction and confidence in prediction. In spite of the fact that our cognitive abilities are well suited to perform high-level analysis and synthesis of ideas around almost any well-observed phenomena of great complexity. We see that when we go deep enough to consideration of fundamental building blocks and basics constituting emerging results of complex social instantiation, our intuitive-based preliminary study of subject with bare mind fails, leaving us with clear understanding of computational infeasibility or false predictions based on fallacies and cognitive miser (Daniel Kahneman, 2002).

That leads to development of scientific approach to reliable study of social phenomena. Such an approach starts with a process of descriptive assessment of observable phenomena and further structuration of it, which yields basic abstractions, followed by axiomatic theories with a lot of assumptions and simplifications entangling those abstractions with each other. Subsequently Economics emerged as a branch of more general social science focusing on study of economic relations, e.g. all those interactions that has something to do with wealth, among people with one of the most influential works called "The Wealth of Nations" (Smith, 1776). Thus, started with macroeconomic theories like those of Circulation in Macroeconomics (Murphy, 1993), AD-AS (Keynes, 1936) economics theory proceeded with microeconomics (Marshall, 1890) and microeconomic foundations of macroeconomics (Lucas, 1976) and postulating that there is might be ambiguous multiplicity of possible predictions (Rizvi, 2006) in context of general market equilibrium. Finally, narrowing the focus of analysis and revision of its assumptions on a quest to better predictions economics theory faces the problem of review of its foundations with regard to individual rationality of economic agents.

That is where Game Theory gains its prodigy, being a mathematics driven approach to formalize strategic-involved interactions between individuals. Strategic-involved in this context means that an individual has a range of possible moves, from which he chooses what his actions would be. Such an approach helped to understand that where is a choice of an individual being involved in the model, there is a range of approaches of how to analyze or predict outcomes of such interactions, variating by assumptions regarding decision-making process of individuals. The most fundamental proposition in this context is a Nash

Equilibrium concept, based on an assumption of mutually known full rationality of decision-making agents; it proved itself as a useful concept for theoretic analysis of social interactions. The most fundamental difficulty with a concept of Nash Equilibrium – is an often seen multiplicity of equilibria. Nash Equilibrium concept and its refinements do not provide any instructions on how to choose over a set of possible equilibria in a task of prediction.

In the same time, Game Theory did not provide a viewpoint on how models under investigation would be actually used in dynamics, one of the most fundamental questions of such kind is “How agents actually choose one equilibrium over another?” With such an approach, it is hard to make reliable practical predictions, so work was done on study of possible dynamics of actual interactions and ways of equilibrium to emerge. There came up a branch of Game Theory called Evolutionary Game Theory, focused on dynamic of strategic interaction. Started with studies in biology, soon it was applied to social science and economics in particular. Nowadays evolutionary economics provides us with great a theoretical foundation of evolutionary perspective on social norms and institutes, and also markets and economy (Nelson, 2015). The great thing about dynamical model of equilibrium convergence is that it makes possible to make a prediction to which outcome model would come based on its current state. Also, evolutionary economics tradition including evolutionary game theory turned out to be much less demanding to cognitive abilities of agents in models, which made it possible apply this concepts to predict experimental results of social studies (Moshe Hoffman, 2005). Therefore, with evolutionary game theory, we got a useful tool to analyze effectively individual rationality or formation of preferences of actions on an individual economic-agent level; it could be referred to Cournot competition model as a historical example of application of game-theoretic concepts (Morrison, 1998).

We mentioned earlier that in a quest for better predictions of social phenomena, people was driven to refine abstractions and related assumptions constituting social models. That led us from macroeconomics to microeconomics to individual rationality scale of analysis. As it was done with micro-foundations of macroeconomics, findings of study of individual rationality should be incorporated as a foundation in larger scale models. Consequently, agent-based models are gaining their popularity in economics analyses. One of the contributing factors to this is a technological progress resulting in a higher availability of computational resources. The fact is that computational complexity of such models makes its analytical analysis unfeasible, therefore demanding computer simulations to be used. Computational complexity of such models is driven by their design, which is agent-centered and expected to derive complex phenomena based on interactions of agents.

Agent-based models depend on many parameters that usually tweaked with respect to the factual observations of process being modeled. Being agent-centered, ABMs have to set some rules by which agents would behave. Taking one or another approach to do so, ABMs are subject to critique of rigidity of assumptions. Moreover, ABMs dealing with social interactions faces the same multiplicity of outcomes as Game Theory. Therefore, we still need some efficient way to investigate possible outcomes.

In this paper, we propose that if there are multiple equilibria in the model of social phenomena being investigated, then it is desirable to be able to make plausible predictions of its dynamics and have an opportunity to influence it. Mechanism Design Theory is a kind of inverse Game Theory. It focuses on deriving rules and structures of social interaction with outcomes having desirable properties. In a process of designing complex mechanism, analysis of its outcomes is hard task because of its computational complexity. So mechanism design could be supported by ABMs (Iris Lorscheid, 2017). In a broad perspective, every ABM involving strategy set of actions available to agents to choose from could be interpreted as a complex game-theoretic model. In this case it is desired to have a reliable way to assess possible outcomes of the model; e.g. to locate multiple equilibria in game-theoretic model and to predict the dynamics of its emergence.

To summarize this part, it could be said that the problem of finding a reliable way with regard to its predictive power to analyze models of social interactions with multiple equilibria could be further decomposed into two sub problems. First – to locate multiple equilibria (outcomes of interaction) and second – to predict dynamic of the model (path of instantiation of a social interaction on the way to predicted outcome) or to predict to which outcome would a system converge from a given initial state. This problem is of a great importance in social sciences in general because its solution allows refining assumptions of outcomes of interactions of social agents in models of complex social phenomena. Looking back to origins of social science, this provides a support to predictive analysis capable to explore possibilities of theoretically justified redesigning the social systems of great complexity such as transportation system (traffic assignment), auctions, taxation, market regulation and other.

The general structure of the paper goes as follow: in this chapter – Introduction – the context and applications of the work are discussed, being finalized by a brief summary of the proposed approach to numerical analysis of game-theoretic models. The next chapter – Literature Review – highlightы findings of prior research on the topic and outlines its background. Then – Model and Algorithm Specification– chapter describes the proposed approach to deal with social-based models and outlines its implementation in an algorithm with pseudocode. The following – Algorithm Assessment– chapter discusses properties of the algorithm and demonstrates a showcase of benchmarks set of synthetic game-theoretic models and generalizes results. The final chapter – Conclusions and Further Research – summarizes obtained results, discusses the approach's applicability to real-world problems and outlines promising directions of further work on the topic.

6 Literature Review

Tackling the problem of locating multiple equilibria of game-theoretic models could be started with a simpler task of locating at least one equilibria. There have been done a lot of work in that field. Eventually a whole branch of algorithmic game theory emerged focused on computational complexity of the problem and different propositions to deal with it (Noam Nisan, 2007). In the same time, a lot of work was done on creating analytical-numerical algorithms to calculate equilibrium points (Paul Frihauf, 2012) (Richard D McKelvey, 1996) (Srihari Govindan, 2004) (Jacek B. Krawczyk, 2000) (Neely, 2013) (Milos S. Stankovic, 2012), for a broad overview of works on finite games please refer to (Stengel, 2010). In addition, there are prior works on application of the approach of evolutionary game theory and evolutionary optimization algorithms to the same task (Mattheos K. Protopapas, 2010) (Rodica Lung, 2007) (Wei-Kai Lin, 2009) (I.A.Ismail, 2007) (Olivier Bournez, 2013) (Veisi, 2012) (Marks, 2002). There are also some works on general-purpose genetic algorithms using concepts from game theory, called Nash Genetic Algorithms, these algorithms are primarily applied to solving engineering problems (Suheyla Ozyildirim, 2000) (Jian Chi, 2012) (M. Sefrioui, 2000) (Kwee-Bo Sim, 2004).

While dealing with small subsets of game-theoretic models, e.g. 2 players' bimatrix games, one would find a great amount of relatively computationally efficient equilibrium locating algorithms yielding one sample equilibrium per run, although not all of them currently implemented as a ready-to-use software packages. Algorithms locating all Nash Equilibria in a given game are constructed around the naïve idea of utilizing formal definition of NE and use systems of polynomial equalities and inequalities based on game's payoffs structure. There is a major disadvantages of such approach: such systems of polynomial inequalities scale up fast with game size (number of players, strategies involved), quickly becoming infeasible to solve (Richard D McKelvey, 1996).

Complexity of the problem of computing equilibria in a loosely specified game is at least PLS-hard problem (solvable by local search) and most plausibly is actually an NP-hard (Roughgarden, 2010). Which means there is no currently an algorithm that guarantees complete solution to the problem in a time shorter than brute-force search through problem space. This finding gives justification to use of algorithms yielding good enough approximations of solution.

In this paper, we focus on using evolutionary optimization approach to tackle this problem. There is a long tradition in application of evolutionary algorithms in the field of Game Theory (Marks, 2002). Starting with works of Axelrod on repeated prisoners' dilemma (Axelrod, 1987). In addition, some works confirm efficiency of evolutionary optimization algorithms in the task of search for equilibria (Rodica Lung, 2007) (I.A.Ismail, 2007) (Wei-Kai Lin, 2009) (Veisi, 2012) (Mattheos K. Protopapas, 2010). These works utilizes a coevolutionary approach and despite the fact that proposed algorithms' performance hugely depends on used parameters, meta-optimization of these parameters is not being done. Moreover, performance of these algorithms measured in non-comparable statistics, such as number of iterations of algorithm being used, that could not been directly related to time or computational resources usage, therefore we incorporate in our work empirical analysis of the influence of parameters of proposed algorithm on its convergence dynamic.

We also want to highlight findings of previous researches on the topic.

There is a branch of works using Lyapunov function to convert the problem of locating multiple equilibria in noncooperative finite games to the problem of locating multiple global minima of single-valued function, as described in (Richard D McKelvey, 1996). In this field (Rodica Lung, 2007) demonstrated successful application of different types of multi-modal optimization algorithms to the task of locating multiple equilibria of game-theoretic models, basically to normal form games, using Roaming Optimization niching technique to preserve multiple equilibria in the population. The very same approach was used in (Veisi, 2012) with similar results. This method requires knowledge about payoff function form and yields all Nash Equilibria without insights of corresponding dynamics of the model, so there are no means to refine found equilibria.

Another branch of work actually uses evolutionary algorithms to work with models without additional transformation. It gives insights on the requirements to the algorithm to be able to locate equilibria of game-theoretic model. In (Mattheos K. Protopapas, 2010) it is shown that evolutionary algorithms are capable to produce Markov Chains as solutions to noncooperative finite games. One of the most fruitful works on application of genetic algorithms to game-theoretic models is (Wei-Kai Lin, 2009). In the paper, they explore applicability of single-population coevolutionary algorithm to locate equilibria of finite noncooperative two-player symmetrical games. Their results give important insights on limitations of using evolutionary algorithms in game-theoretic context and characteristics of equilibria subset that could be found with them, such as the fact that not every mixed strategy could be located.

7 Model and Algorithm Specification

In the paper, we try to come up with a general-purpose algorithm to investigate social models with regard to possible multiplicity of equilibria. Therefore, we want our algorithm to possess following properties: anytime algorithm, freedom of model specification, capable of handling multiple equilibria, insightful on dynamics of the model. To do this we start up with a formalized definition of social interaction or game in which we are going to search for equilibria. Then we define what would be treated as equilibria or stable state of the model and show how a model could converge to one or another equilibrium from an arbitrary state. After that, we propose a numeric algorithm to tackle this model, inspired by evolutionary optimization algorithms and possessing all desired properties listed up before.

7.1 Model of Social Interaction

7.1.1 Game

Our work focuses on a concept being called ***social interaction*** or ***game***, which is finite, e.g. final scores or payoffs or utilities of the players (agents being involved), could be derived. General structure of the game consists of:

- a set of agents
- for each agent, a set of actions
- payoff function that assigns a payoff to each agent

$$\Gamma = (\mathrm{A}, \Delta, \Psi)$$

Γ - game

A - agents set

Δ - actions set

Ψ - payoff function

Game goes as follows: agents choose their actions simultaneously, and then payoffs are calculated. Our model could be treated as a non-cooperative model, because it does not take into account possible coalitions between agents explicitly. We favoring indeed specification of cooperation involved models in non-cooperative terms, because decision to participate or not in a particular coalition is taken by each agent independently, or being a result of combination of individuals' independent decisions.

7.1.2 Agent Set

Agent set A is a finite set of size n that serves as a list of agents being considered in the model.

$$\mathrm{A} = \{a_1, \dots, a_n\}, |\mathrm{A}| = n$$

7.1.3 Action Set

Actions set A is a compact space of all possible actions that agents could undertake in the model. Action set could be divided into subsets corresponding one-to-one to agents – agents' decision-spaces, representing all possibilities of actions available to a given player. Then each decision space could be further broken down to decision-sets that could be either discrete sets or compact spaces, representing for a given agent available choices in a particular decision she makes.

For example, in some model there could be 10 firms (agents) each having 2 decision-sets: first – what product out of 5 possible products to produce (discrete decision-set of 5 points), second – what proportion of taxes should it pay to government (continuous compact decision-set). In this case, action set of the model would consist of 10 decision-spaces (one for each agent) each including 2 decision-sets – discrete and continuous – 20 decision-sets in total.

$$\Delta = \{D_1, \dots, D_n\},$$

D_i - decision-space;

$$\Delta \cong \mathrm{A};\ |\Delta| = |\mathrm{A}| = n;$$
$$D_i \neq \emptyset, \qquad D_i = \{d_{i_1}, \dots, d_{i_m}\};$$
d_{i_j} - decision-set
$$\forall d_{i_j}: d_{i_j}\ is\ discrete\ set\ OR\ compact\ space$$

For some models it could be desirable to have unbounded continuous decision-sets, for example, in a model of price competition price set by a firm might lay in unbounded interval $[0, +\infty)$. In such case there are two possible outcomes of locating equilibrium: equilibrium price exists, therefore its finite value could be enclosed in compact space, or there is no equilibrium price; in the second case agents might be better off rising prices infinitely. With respect to practical application of the model, we propose that such price interval could be closed with upper bound sufficiently large enough to include any reasonable price that might be found in assumed equilibrium. Then, dynamics of the model in the case when there is no equilibrium and agents better off rising prices infinitely could be easily detected by preferred prices approaching to upper bound of price interval.

7.1.4 Payoff Function

Payoff function Ψ takes a point from Cartesian product of decision-sets contained in Action set and returns a vector of payoffs of the same cardinality as Agent set, e.g. containing payoff for each agent. Every payoff of an agent is a real value.

$$\Psi: \prod_{d \in \Delta} d \to P;$$
$$P \cong \mathrm{A};\ |P| = |\mathrm{A}| = n;$$
$$P = \{p_1, \dots, p_n\}, p_i \in \mathbb{R}$$

It is assumed that agents has simple preferences over payoffs and payoffs satisfies four axioms of Von Neumann–Morgenstern utility theorem (John von Neumann, 1944), thus $p_i \prec p_i' \Leftrightarrow p_i < {p_i}'$.

Strictly speaking, we allow Ψ being not only a function, but also an operator of obtaining an outcome in terms of probability theory, allowing for noise in payoffs and random variables involved in payoffs calculation.

7.2 Equilibrium Definition

We use the Δ_i notation to denote the subset of action set that consist of actions related to agent i, and Δ_{-i} notation to denote the subset of action set consisting of actions of all other agents. This notation is used with other sets in the same manner, where it makes sense.

7.2.1 Mixed Strategies

In our model, we assume that agents could choose their actions by any rules. We represent such rules for each agent by a set of probability distributions over decision-sets of the agent, each of which being called strategy. By this, we incorporate mixed strategies in our model.

$\forall d \in \Delta, \exists \phi: \phi$ is a random variable with probability distribution over d:

$\forall \phi: \phi \in \Phi$

$\Phi_i \coloneqq set\ of\ \phi\ corresponding\ to\ d \in D_i, |\Phi_i| = |D_i|$

7.2.2 Agents' Optimization Problem

We propose that agents want to maximize their expected payoffs by tweaking their strategies. Therefore, each agent solves the following optimization problem:

$$\forall a_i \in \mathrm{A}: \Phi_i = \arg\max_{\Phi_\mathrm{i}} \mathbb{E}[p_i \in \Psi(\Phi_i \times \Delta_{-i})]\, , given\ \Delta_i\ constant.$$

7.2.3 Stable State

We define a stable state (equilibrium) in the model as a state in which no agent could increase her expected payoff, given others strategies stable. At least one such point always exists, because in any game-theoretic model exists a Nash Equilibrium in which this condition holds by definition (Nash, 1951).

$$\forall a \in \mathrm{A}: \exists \Phi_a : \mathbb{E}[p_a \in \Psi(\Phi_a \times \Delta_{-a})] \geq \mathbb{E}[p_a \in \Psi((\Phi_a)' \times \Delta_{-a})]$$

7.2.4 Optimization Dynamic

We assume agents to change their strategies by small perturbations with a drift towards strategies yielding higher expected payoff until there would be no possible perturbations yielding higher expected payoff than current strategy yields. In this sense, agents in our model always use totally mixed strategies, because of non-zero chance for each agent of perturbation her strategy to a strategy picking a previously unused action with non-zero probability.

7.3 Algorithm Specification

7.3.1 General Evolutionary Algorithm Structure

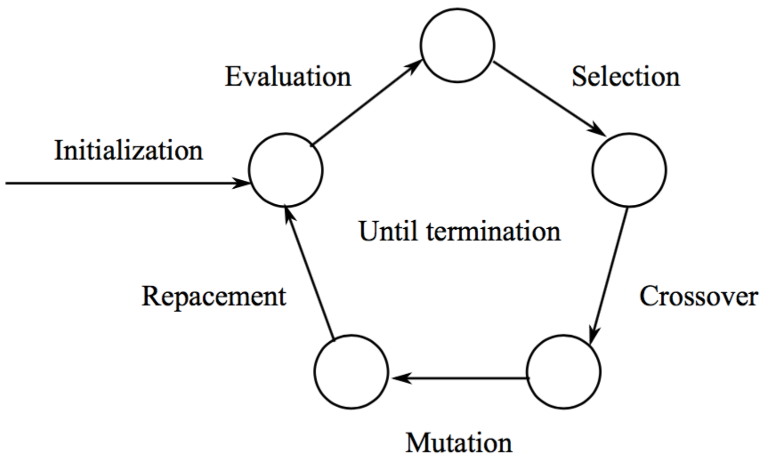


Figure 1: Flowchart diagram of an arbitrary genetic algorithm. Adapted from (Wei-Kai Lin, 2009, p. 8).

Evolutionary Algorithm (EA) is a large class of black-box optimization routines inspired by Darwinian evolution process. Attributes of evolutionary systems being utilized in evolutionary algorithms are (adopted from (Jong, 2006 p. 2)):

- one or more populations of individuals competing for limited resources,

- the notion of dynamically changing populations due to the birth and death of individuals,
- a concept of fitness which reflects the ability of an individual to survive and reproduce, and
- a concept of variational inheritance: offspring closely resemble their parents, but are not identical

Genetic Algorithm (GA) is a subclass of EA, tradition of implementation of outlined above principles. Its advantage is in universality and it has proved itself as a good-performing search algorithm on many problems. GA uses string representations of solutions in the search space, each such representation called *individual* or *specie*. A set of such species presented in a given iteration called *population*, iterations called *generation*. To differentiate species in a population *fitness valuation* is used, for example in a task of function maximization a value of a function in a point represented by an individual could be used as a fitness value. The general idea of GA is to evolve populations from generation to generation favoring in transition between generations individual with highest fitness value, concurrently maintaining diversity of population. Conceptually GA consists of a few basic steps; we will use Figure 1 as a reference and describe each step presented on the figure clockwise.

1. ***Initialization***. Algorithm initialized by creating a population uniformly distributed across search space.
2. ***Evaluation***. Each specie is being assigned a fitness value based on its performance on the target function.
3. ***Selection***. Subset of population is selected by some rules favoring highest fit individuals to be used later to produce offspring and/or to be transferred to next generation.
4. ***Crossover***. Selected individuals produce offspring by combination of their traits, for example in the case of real-valued vector encoding of points in search space simple averaging of their components might be used.

5. ***Mutation***. Selected individuals (including offspring) change some of their traits in a predefined random manner to explore solution space and maintain diversity of population.
6. ***Replacement***. Population of the new generation composed. It is usually preferred to maintain population size of each generation on the same level, so parents (members of the previous generation) and offspring (potential members of the new generation) should be mixed together in some way to form new generation.

7.3.2 Algorithm Outline

Here we present an algorithm proposed to locate equilibria of game-theoretic models. We start with a stochastic algorithm of GA type, capable of locating one equilibrium per run, depending on initial seed of the algorithm. For locating multiple equilibria, we use repeated local search, e.g. running algorithm multiple times with different initial seeds, as a robust naïve approach to multi-modal optimization (Preuss, 2015 pp. 59-60).

As an input for the algorithm, we expect a game model as it outlined in Model of Social Interaction section of the paper. We want to highlight the fact that for algorithm to work it is enough to know just the structure of action set Δ (agents and their decision-sets) and an interface to compute corresponding payoffs (there is no need actually to know how these payoffs are calculated). As an output, we expect a found equilibrium or statement of failing to find one, in either case output should contain log of convergence dynamic, which consists of individuals' encodings in each generation and their fitness. This algorithm could be started several times to find multiple equilibria.

We do not discuss in this chapter criteria of determining if equilibrium is found, because it depends on the particular type of model being investigated. Due to stochastic nature of the algorithm and noise in payoffs, every convergence criteria would be based on some heuristics, therefore the most reliable method of determining if algorithm converged would be manual inspection of its convergence dynamics.

Throughout this section, we use terminology defined in previous sections of Model and Algorithm Specification chapter.

7.3.2.1 Encoding

In this work, we utilize traditionally optimization algorithm to locate equilibria of game-theoretic models. Therefore, it is important as a first step to define target function, search space and its encoding for the algorithm.

We will start with definition of search space. Search space being investigated by the algorithm is conjecture of all possible probability distributions (mixed strategies) over decision-sets in provided model, which collective outcome lies in a Cartesian product of all decision-sets in action set $\prod_{d\in\Delta} d$. Each point of which corresponds to a particular set of actions picked by agents and could be used as an input for payoff function Ψ. Therefore, the full encoding of a solution consists of encodings of strategies on every decision-set in the model.

As mentioned earlier decision-sets could be of two types: discrete or continuous. In this paper, we restrict our analysis to the models containing only discrete decision-sets. Probability distributions over discrete decision-sets could be numerically encoded straightforwardly by assigning real-valued non-negative probabilities to points of decision-set, which in sum should be equal to one. In the case of continuous decision-sets, continuous probability distributions over them could be approximated by discrete probability distributions (Ken'ichiro Tanakaa, 2012).

7.3.2.2 Structure of Populations – Coevolution

Usually in GA in every generation just one, single, uniform population of individuals representing solutions in a search space is maintained. Each specie has full encoding of a solution in the search space. To assign fitness value to an individual its encoding of a solution used to calculate value of the function being optimized and its value then used to calculate the individual's fitness. This works fine when the function being optimized single-valued or could be transformed to single-valued function. However, it is not the case with game-theoretic models where payoffs of agents could not be aggregated in a meaningful way maintaining independence of strategic interests of agents.

Thus, we encode strategies on different decision-sets independently and keep individuals encoding different parts of full solution encoding in distinct subpopulations, therefore we maintain as many subpopulations as decision-sets presented in a model. Consequently, a single member of a subpopulation could not be evaluated without matching with some members of other subpopulations, because it is needed to provide a complete point in Cartesian product of decision-sets to the payoff function to derive individual's payoff and assign fitness. Below we provide an illustration of possible generation structure of the algorithm dealing with a sequential form of the game "Battle of Sexes".

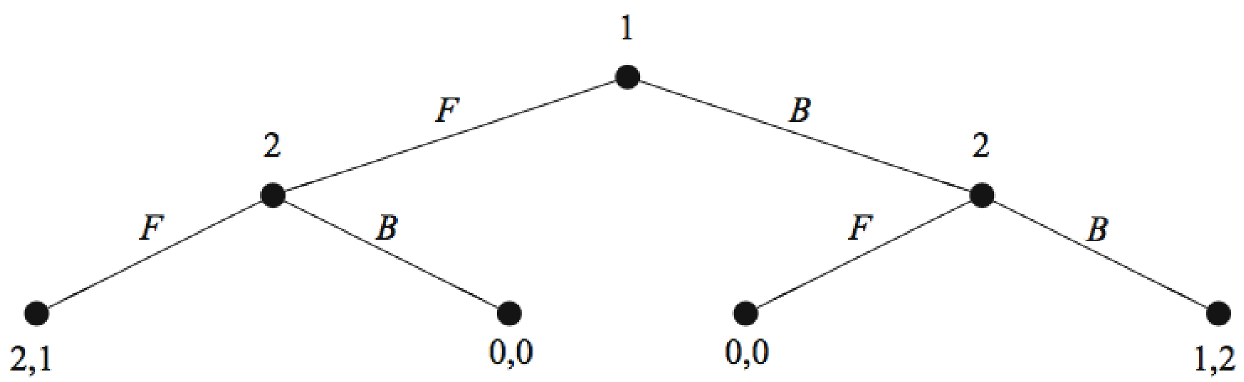


Figure 2: The decision tree of sequential battle of the sexes.
Adopted from (Peters, 2015 p. 8)

Generation ###		
subpopulation 1 Player 1 F / B	subpopulation 2 Player 2 F: F / B	subpopulation 3 Player 2 B: F / B
1.000; 0.000	1.000; 0.000	0.000; 1.000
0.991; 0.009	1.000; 0.000	0.003; 0.997
0.975; 0.025	0.989; 0.011	0.000; 1.000
0.992; 0.008	0.995; 0.005	0.000; 1.000

Figure 3: Example of the structure of a generation of proposed algorithm for sequential form of BoS game.

7.3.2.3 Fitness Evaluation – Matching Algorithm

To evaluate specie's fitness from a subpopulation its encoding of solution must be completed using encodings of one specie out of each other subpopulation. So one needs to match species between subpopulations to evaluate their fitness. Failing to maintain an exchange of encodings between subpopulations results in algorithm's non-convergence (Mattheos K. Protopapas, 2010).

There are two general approaches to do that in practice: asynchronous and synchronized. In asynchronous evaluation each subpopulation evolve on its own, periodically communicating with other subpopulations to exchange information about reference individuals used to derive complete encodings of solution (Bull, 2001). In synchronized evaluation for each evaluation of an individual of a subpopulation, the individuals from other subpopulations picked at random (Fogarty, 1993). For the implementation of the algorithm considered in this paper, synchronized evaluation is chosen over asynchronous one because of its simplicity and ease of interpretation in context of randomness involved in calculation of fitness value. It should be noted that actual implementation of matching algorithm hugely influences convergence dynamic of an algorithm, as depicted in (Husbands, 1994).

We propose a matching algorithm that takes an amount of desired matchings to be produced (same for each individual in the population) as an input $NPAIRS \geq 1$ and returns a list of $NPAIRS * POPSIZE$ matchings (set of individuals, one from each subpopulation, collectively bearing a full encoding of a solution for a model); it is assumed that each subpopulation has the same size of $POPSIZE$. We use $\mathfrak{d}_i$ to denote a subpopulation corresponding to the decision set d_i; collection of all subpopulations denoted by $\mathfrak{D}$; members of subpopulations are strategies over corresponding decision-set denoted by ϕ. For reference, check Figure 3: Example of the structure of a generation of proposed algorithm for sequential form of BoS game. Here is an outline of the matching algorithm – Algorithm 1 – in pseudocode:

1. Take input $NPAIRS \geq 1$
2. Initialize list of matchings $\mathbb{M} = \emptyset$
3. Make lists $\mathfrak{d}'$ of members of each subpopulation $\mathfrak{d}$
4. For each member of each subpopulation
    a. initialize counter $\rho = 0$
5. Initialize new matching $\mathbb{m} = \emptyset$
6. For each list $\mathfrak{d}'$
    a. Take a random individual i from $\mathfrak{d}'$
    b. Add i it to the matching $\mathbb{m}$
    c. Increment individual's counter by one $\rho_i += 1$
    d. If $\rho_i = NPAIRS$, then exclude individual i from list $\mathfrak{d}'$
7. Add matching $\mathbb{m}$ to list of matching $\mathbb{M}$
8. If any $\mathfrak{d}' \neq \emptyset$, then go to step 5
9. Return list $\mathbb{M}$

Algorithm 1: Matching Algorithm

Finally, to assign fitness values we take matchings list, for each strategy in a matching, we sample an action; and then we derive payoff function values and increase each matching strategy fitness value by a payoff of the player to which the strategy belongs. This procedure could be repeated several times for the same matching; in the algorithm and further in the paper we refer to number of times each matching being evaluated as a parameter $NGAMES$. While the algorithm handling mixed strategies, each evaluation of payoff function gives payoffs for sample pure strategies. However, equilibrium or stable states defined in terms of expected payoffs from mixed strategies. In this context, compounding multiple evaluations of fitness value of the same strategy is desirable, with averaged fitness value converging to true expected payoff as a number of valuations goes to infinity.

7.3.2.4 Selection – New Generations Formation

For implementation of our algorithm, we used truncation selection procedure applied to each subpopulation independently, which in absence of crossover operations fully determines the dynamics of construction of new generations. Truncation selection is an algorithm with just one parameter $DROPOUT\_RATE$, which specify as a proportion $(0; 1)$ what part of current generation's least fitted individuals would be dropped out and replaced by offspring of more fitted individuals. Selection goes as follows: population members sorted by fitness, then $\lfloor POPSIZE * DROPOUT\_RATE \rfloor$ (but at least one) least fitted individuals are deleted from population and the same amount of offspring of more fitted individuals is produced and added to population.

Offspring of the most fitted individuals in a population (those left after truncation) generated by random selection of individuals in amount needed to replace dropped individuals, duplicating it and applying mutation operator to duplicates.

Truncation selection is chosen for implementation because of its simplicity and robustness in conjunction with mutation operator. For the overview of possible selection procedures and discussion of their efficiency please refer to (Sevan G. Ficici, 2005).

7.3.2.5 Mutation

In proposed algorithm mutation is the only operation that preserves diversity of population and provides search space exploration means for the algorithm.

Encodings of solutions in the algorithm consists of real numbers in closed intervals. Hence we construct a mutation operator as function adding to each component of encoding being mutated a sample of random variable normally distributed with mean of zero and standard deviation equals to a given percentage of component's boundary interval length, followed by normalization of encoding to be in alignment with it boundaries. Therefore, mutation operator also takes just one parameter $MUTATION\_MAGNITUDE$ in a range of $(0; 1]$.

For example, let us consider a case of a strategy over two points discrete decision set being mutated, while $\mathrm{MUTATION\_MAGNITUDE}$ parameter is set to 0.03. Let us assume that it current encoding is $\{0.5, 0.5\}$. Its encoding consists of two real numbers, so to mutate it we are going to use two sample of random variable with normal distribution with mean 0 and standard deviation $(1 - 0) * 0.03 = 0.03$. So sample obtained and equals to $\{0.184, -0.052\}$. We add it to current encoding and obtain $\{0.684, 0.448\}$ which doesn't sum up to unity $0.684 + 0.448 = 1.132 \neq 1$, thus it cannot be used to encode discrete distribution. To fix that we normalize components of our encoding as follows $\left\{\frac{0.684}{1.132}; \frac{0.448}{1.132}\right\} = \{0.604; 0.396\}$, which sums up to 1 and legible to encode discrete probability distribution.

7.3.2.6 General Structure of the Algorithm

Here we summarize proposed algorithm – Algorithm 2 – in pseudo code:

1. Take $GAME\_MODEL$ as input with parameters
2. Generate an initial subpopulation of uniformly distributed in search space individuals for each decision-set in the model
3. Evaluate fitness for each individual
4. Apply truncating selection and mutation procedures for each subpopulation
5. Construct new generation of subpopulations
6. If termination condition is not met, then go to step 3
7. If equilibrium is found, return found equilibrium
8. Return log of convergence dynamic

Algorithm 2: General Structure of the Algorithm

8 Algorithm Assessment

In this chapter, we are going to investigate performance of the proposed algorithm empirically by assessing its performance on selected test problems. All source code that was used to obtain further results would be attached to the paper for reproducibility of results. In addition, we are going to discuss some findings on characteristics of equilibria that could be found by the algorithm. We will finish this chapter by Monte Carlo experiments on applicability of the algorithm as is to the problem of locating multiple equilibria of game-theoretic models.

8.1 Note on Implementation Design

The Genetic Algorithm nature of the proposed algorithm possess one very pleasing property – it is "embarrassingly parallel" (Cantú-Paz, 2001). Which means that computations of the algorithm could be separated into parallel tasks naturally, because there is no interdependency between them. Indeed every evaluation of fitness values for each matching could be done separately from other computations. Results of these computations need to be collected just once to apply the selection operator to population. Working on a task of locating multiple equilibria with current approach of repeated local search (multiple runs of the algorithm with different initial seeds), all runs of the algorithm could be done in parallel.

This paper focus on developing an algorithm that could be used as a practical tool to locate multiple-equilibria in complex models of social interactions, i.e. agent-based models of social systems. We anticipate the fact that a single evaluation of such model might take a considerable amount of time. However, effective leveraging of the parallel nature of the algorithm allows keeping running time of the algorithm relatively low. To illustrate this let us assume that a single evaluation of a model in which equilibria are searched takes $\mathfrak{T}$ amount of time, for simplicity we also assume that it doesn't depend on parameters of the model, that is average running time of evaluation equals worst-case running time. It could also assumed that in real application $\mathfrak{T}$ would be significantly larger than overhead operations of the algorithm to manage generations. So for example, we want to run the algorithm 100 times to get a sample of possible equilibria and each time algorithm would run for 30 generations. Since 100 runs of the algorithm could be done in parallel, it will take the same time as just one run if all runs are computed in parallel. Consequently, a single run of the algorithm would evaluate fitness for 30 generations of strategies. As it was said earlier, all fitness evaluations inside a generation could be done in parallel, so with constant running time of a single evaluation, fitness evaluation for the whole generation takes as much time as it takes to perform a single fitness evaluation – in our case it is $\mathfrak{T}$. As a result, running algorithm for 100 times, each for 30 generations, with possibly many model evaluations for each generation would take us only $30 * \mathfrak{T}$ of running time.

Because of the reasoning outlined above in further research we favors fast (in terms of number of generations needed to be evaluated) convergence of the algorithm.

8.2 General Convergence

8.2.1 Setup of the Experiment

We will start by benchmarking our script on the well-known game “Prisoners’ Dilemma” Figure 4.

<table>
<tr><td colspan="2"></td><td colspan="2">Player 2</td></tr>
<tr><td colspan="2"></td><td>Cooperate</td><td>Defect</td></tr>
<tr><td rowspan="2">Player 1</td><td>Coopera</td><td>-1; -1</td><td>-3; 0</td></tr>
<tr><td>Defect</td><td>0; -3</td><td>-2; -2</td></tr>
</table>

Figure 4: Prisoners' Dilemma payoffs matrix

We want to investigate influence of parameters on convergence of the algorithm. To do that we would construct a mesh grid of parameters and evaluate each combination of them several times.

Let us remember that the algorithm depends on 5 parameters: $POPSIZE$, $NPAIRS$, $NGAMES$, $DROPOUT\_RATE$, $MUTATION\_MAGNITUDE$. We construct a mesh grid by taking equally distanced points in specified ranges for each parameter independently and then combining them. We use the following parameters to construct the mesh grid (notice that parameters $dropout\_rate$ and $mutation\_magnitude$ represented here as percentages, rather than fractions):

Notation: 'PARAMETER_NAME':(lower bound, upper bound, number of points)

{'popsize': (5, 100, 7), 'npairs': (1, 40, 7),
'ngames': (1, 1, 1), 'dropout_rate': (1, 80, 12),
'mutation_magnitude': (1, 80, 12)}

In total: $7 * 7 * 1 * 12 * 12 = 7056$ combinations

We hold $NGAMES$ at constant level 1, because generally its purpose is to control over precision of expected payoff calculation, for which it could be seamlessly substituted by $NPAIRS$ parameter. It could not be done other way around, because $NPAIRS$ also influence diversity of opponents (or matching encodings) used to evaluate fitness of each individual. That is, on low values of $NPAIRS$, individuals get biased estimates of expected payoffs of their strategies relatively to surrounding population of possible opponents to estimate expected payoffs. We leave empirical investigation of the effect of $NGAMES$ parameter on algorithm performance for further research.

Knowing that the only equilibria of the model is in pure strategies, we propose the following convergence metric: the model converged if and only if for two consecutive generations best fitted individuals of each subpopulation represent the same pure strategies. When model converged, we terminate further calculations of next generations, further in the text we refer to this as a termination condition; also, we terminate calculations after 40 generations without convergence.

We evaluate each combination of parameters 10 times, totaling in 70560 model evaluations.

8.2.2 Processing of Experiment's Data

When computation of the experiment finishes we process obtained results in the following way. For each model run, we keep two datasets. First with metadata of model run including number of generation when the algorithm converged (encoding non-converged runs by convergence on 0 generation), parameters of the run, flag of convergence ('True' if terminated by termination condition, 'False' otherwise) and category of outcome. Category of outcome takes one of 3 possible values: "non-converged" for non-converged runs, "true_equilibrium" for runs where one of true equilibria of the model found, "false_convergence" for runs that was terminated as converged and which returned solution not being valid equilibrium of the model – false-positive convergence).

8.2.3 Results

8.2.3.1 Convergence

First, we construct a histogram showing at which generation the algorithm has converged, encoding non-converged runs as convergence on the 0 generation false-positive . Number of counts of each outcome presented in table Figure 5.

Model Run Outcome	false_convergence	non_converged	true_equilibrium
Count	62	8747	61749
Percentage	$< 0.1\%$	12.4%	87.5%

Figure 5: Test run's on Prisoners' Dilemma outcomes

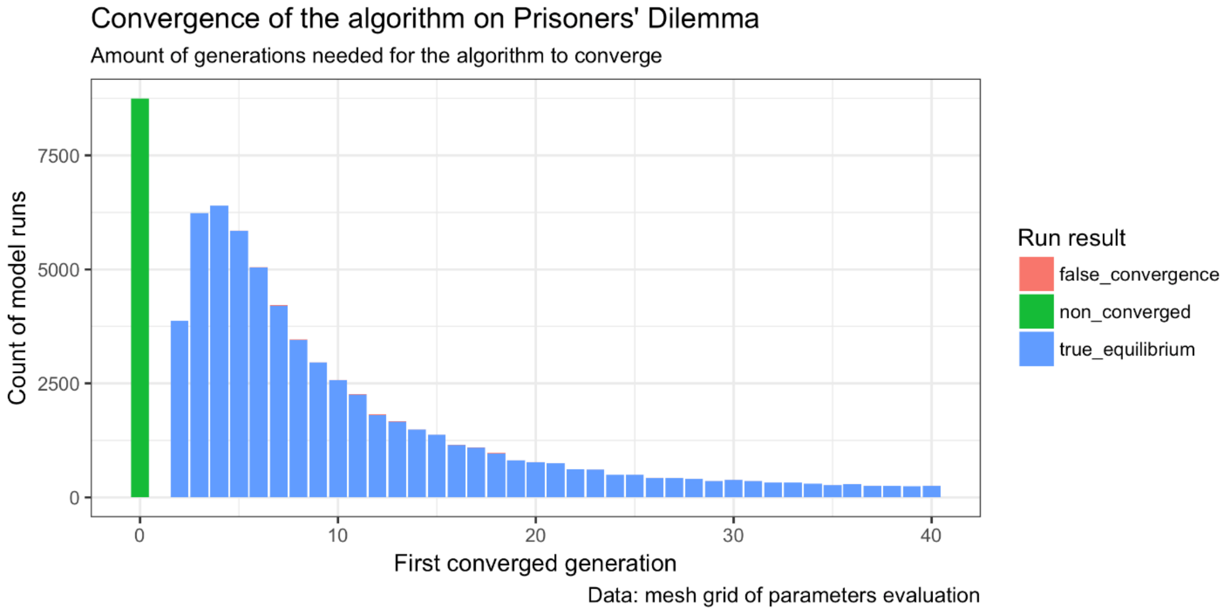


Figure 6: Convergence of the algorithm on Prisoners' Dilemma problem

As it could be seen from the graph Figure 6 with a given mesh grid of parameters model successfully converged in most cases, resulting in nice left-skewed histogram. Also, amount of model runs resulted in convergence to non-legible equilibrium (false-positive convergence) could be taken as negligible.

8.2.3.2 Running Time

As it was discussed earlier, we favor algorithm convergence in short amount of generations. To foster faster convergence through exploitation of the parallel nature of the algorithm we increase number of model evaluations in each generation to provide better guidance to selection operator and thus increase selection pressure on inferior individuals. Increasing number of fitness evaluations per generations in parallel manner means involving more computational power in number of processors used. In a context of scarcity of computational resources, it means higher expenses per unit of time. Because of this, we present a graph Figure 7 of number of generations needed for model to converge and corresponding average amount of fitness evaluations per generation.

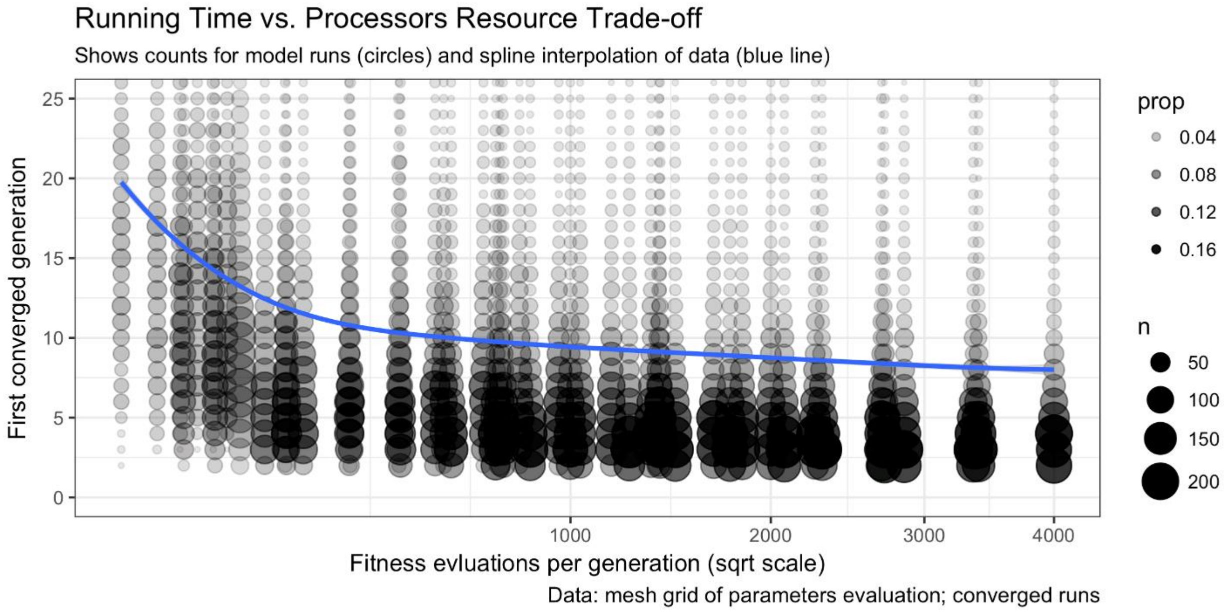


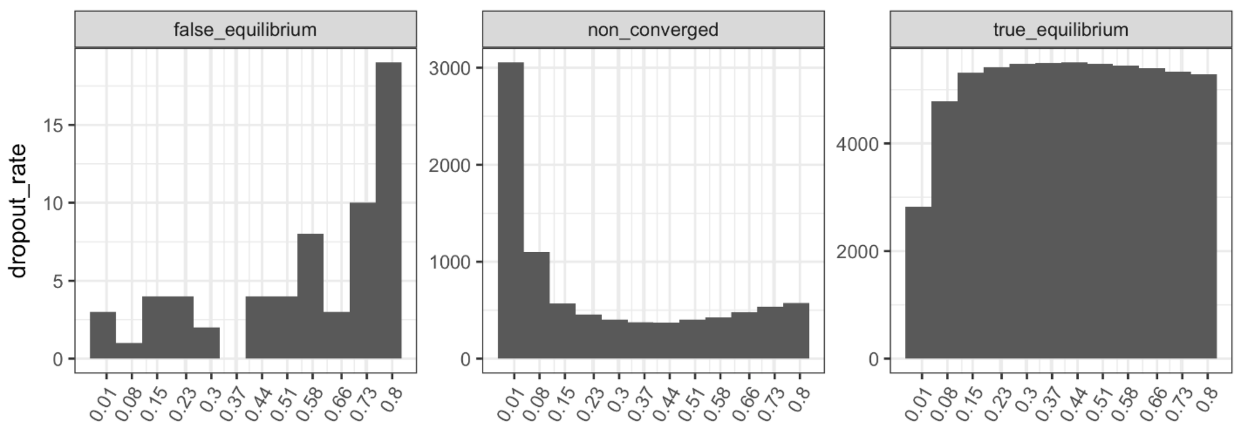
Figure 7: Running Time vs Processors Resource Trade-off

As it could be seen from the graph, aforementioned trade-off between running time and computations needed in each generation has an expected form of isoquant. It also could be seen from this graph that number of fitness evaluations' magnitude of influence on number of generations to convergence diminishes quickly after approximately 700 fitness evaluations per generation.

8.2.3.3 Parameters influence

Now we are going to investigate the influence of parameters of the algorithm on its performance.

We start with investigating of parameters distributions among algorithm runs across different outcome categories.



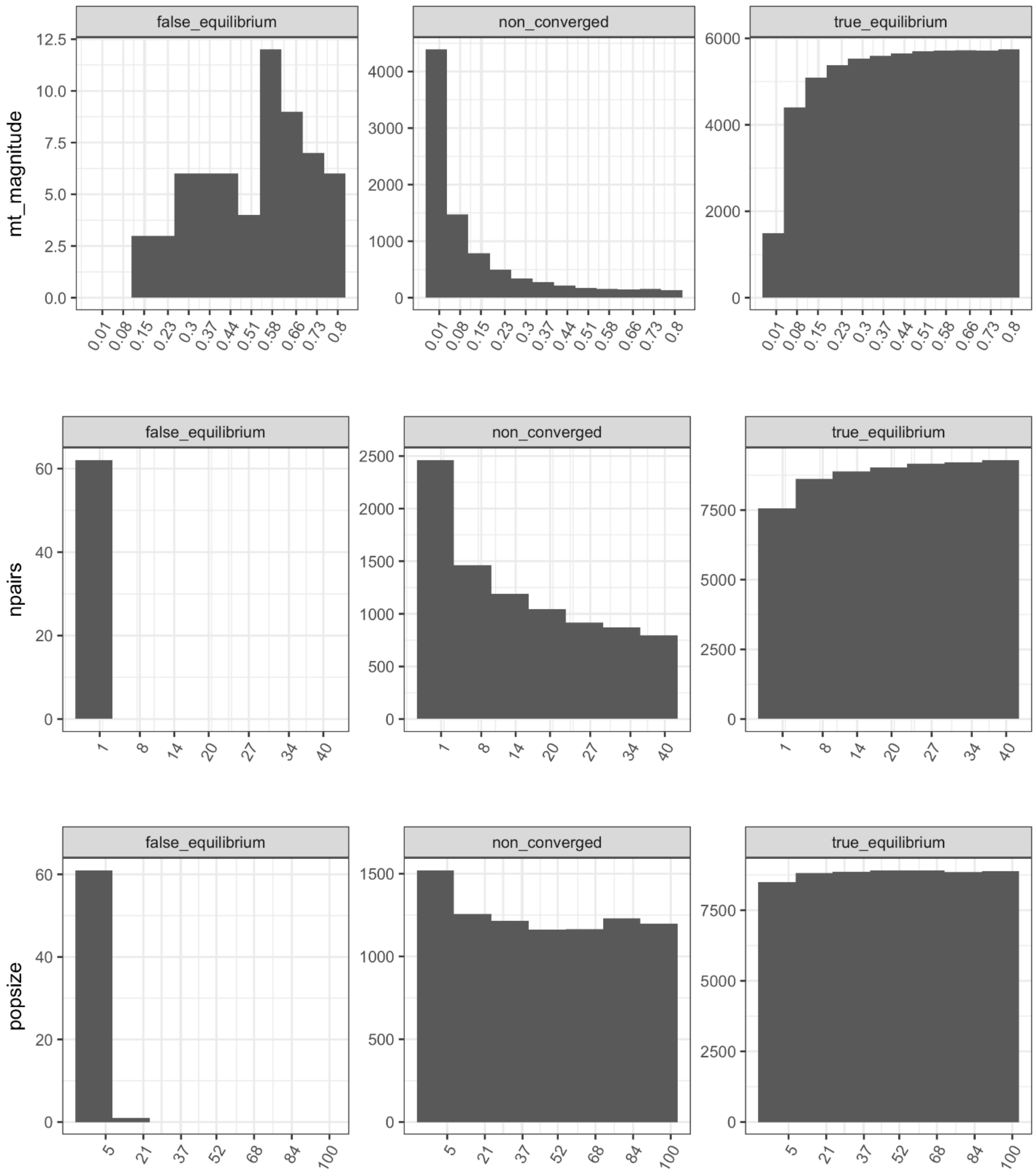


Figure 8: Parameters' influence on algorithm run's outcome

These graphs Figure 8 goes along with intuitive stylized facts regarding influence of parameters on performance of the algorithm.

Dropout rate at low levels makes algorithm is too conservative in exploration of search space, thus the algorithm fails to converge in limited number of generations. It could be said, that with dropout rate reaching 0 number of generations need for the model to converge goes up to infinity – this analysis is valid with an assumption of nearly infinitely large population, because in current design of the algorithm at least one specie is dropped out of population. In the same time proportion of runs converged to point not being an equilibrium of the model is rising along with an increase of dropout rate. With high values of this parameter algorithm does not preserve enough amount of well performing species between generations, hence fails to converge.

Mutation magnitude also defines characteristics of explorative behavior of the algorithm. It could be expected that proportion of non-converged models monotonically decreasing with an increase of mutation magnitude. Proportion of convergences to non-equilibrium points is generally expected to rise monotonically with mutation magnitude, because with higher values of the parameter the algorithm traverse search space more cursory. However, we see different result on empirical data with distribution single peaked around average values of mutation magnitude. It could hypothesized that this effect emerges due to the fact, that higher diversity of population (in terms of variance of strategies) obtained with higher values of the parameters leads to better approximation of expected payoffs of strategies, because every one of them is being evaluated against broader range of complementary encodings of solution space.

Expectations of response dynamics of the algorithm on changes of $NPAIRS$ parameter goes fully along with empirical results. With increase in parameter's value, proportions of true equilibria – rises, false equilibria – declines, non-convergence – declines monotonically. This parameter influences amount of evaluations of each strategy, thus influences precision of approximation of expected payoffs of strategies.

Algorithm's response to changes in $NGAMES$ parameter is expected to has exactly the same dynamic as in the case with $NPAIRS$ parameter.

Algorithm's dynamic relatively to different sizes of population is the same as with values of $NPAIRS$ parameter, although underlying reason is different – size of population influences diversity of population in terms of amount of maintained species.

8.3 Note on Equilibria Concept

8.3.1 Convergence to Evolutionary Stable Strategies

Our algorithm uses truncation selection and mutation as drivers of evolutionary dynamic in process of convergence. Relation of this selection procedure to the classic replicator dynamic (Smith, 1982) was investigated in (Sevan G. Ficici, 2005) and (B. Morsky, 2016). It was shown there that truncation selection leads an evolutionary system to convergence to Evolutionary Stable Strategy.

ESS is solution concept in evolutionary game theory defined for models allowing only pure strategies for agents as stable proportions of pure-strategist individuals with certain strategies in a population, therefore it could be treated as a stable state of population as a whole. In the next chapter, we would discuss thoroughly interpretation of this concept in mixed-strategist models and Mixed ESS and their properties. In this chapter, we restrict our analysis to models with monomorphic ESS, which is ESS that could be represented by a single pure strategy, so by now it enough to be said that every ESS is a Nash Equilibrium.

Aforementioned works on convergence of truncation selection to ESS investigate the subject using classic pure-strategist models, in which individuals restricted to follow just one pre-determined pure strategy throughout their lifecycles. It should be noted that under such conditions in the absence of diversity preserving mechanisms truncation selection under some circumstances might fail to converge, due to extinction of some strategies. However, it is not the case with the algorithm proposed here, since it uses mutation operation to explicitly foster diversity of the population. Hence in the long-run it is converges asymptotically to Evolutionary Stable Strategy (if there any) of processed model by the design.

8.3.2 Multiplicity of Evolutionary Stable Strategies Definitions

Our algorithm converges to ESS. However, in our model we employ mixed-strategist models, which is not widely used in literature on evolutionary game theory and which defines some peculiar properties of system dynamics.

At first it should be denoted that there are two definitions of ESS: the classic one by (Smith, 1982), and the stricter one by (Thomas, 1985). These definitions differ slightly from each other and do not coincide sometime. Due to the lack of prior work on the topic, there is might be an ambiguity to what equilibrium concept does an algorithm as the one proposed here converges. To determine it we arrange an empirical experiment involving a game called “Harm thy Neighbor”. It has two Nash Equilibria in pure strategies [A; A] and [B; B]. In definition of ESS by (Smith, 1982) *both* of NE are ESS. In the same time, in definition of ESS by (Thomas, 1985) *none* of NE is ESS – there is no ESS in this game.

		Player 2	
		A	B
Player 1	A	2; 2	1; 2
	B	2; 1	2; 2

Figure 9: HarmThyNeighbour game's payoffs matrix

To check to ESS of which definition our algorithm converges. We run this game without termination criteria for 2000 generations. Expecting, that if the algorithm converges to the stricter definition of ESS, then it would fail to converge to a stable state in this setup and even if once in a while it would reach a generation with best fitted solution representing one of NE, it would diverge from it and we will see oscillating behavior of the population. Otherwise, if algorithm converges to ESS of classic definition, we would get confident convergence behavior, and a population would stick to one of the equilibrium and actually, the whole population (not just the best-fitted solution) would converge to it.

Algorithm run parameters:

popsize: 54, npairs: 20, ngames: 1, dropout_rate: 37, mutation_magnitude: 8

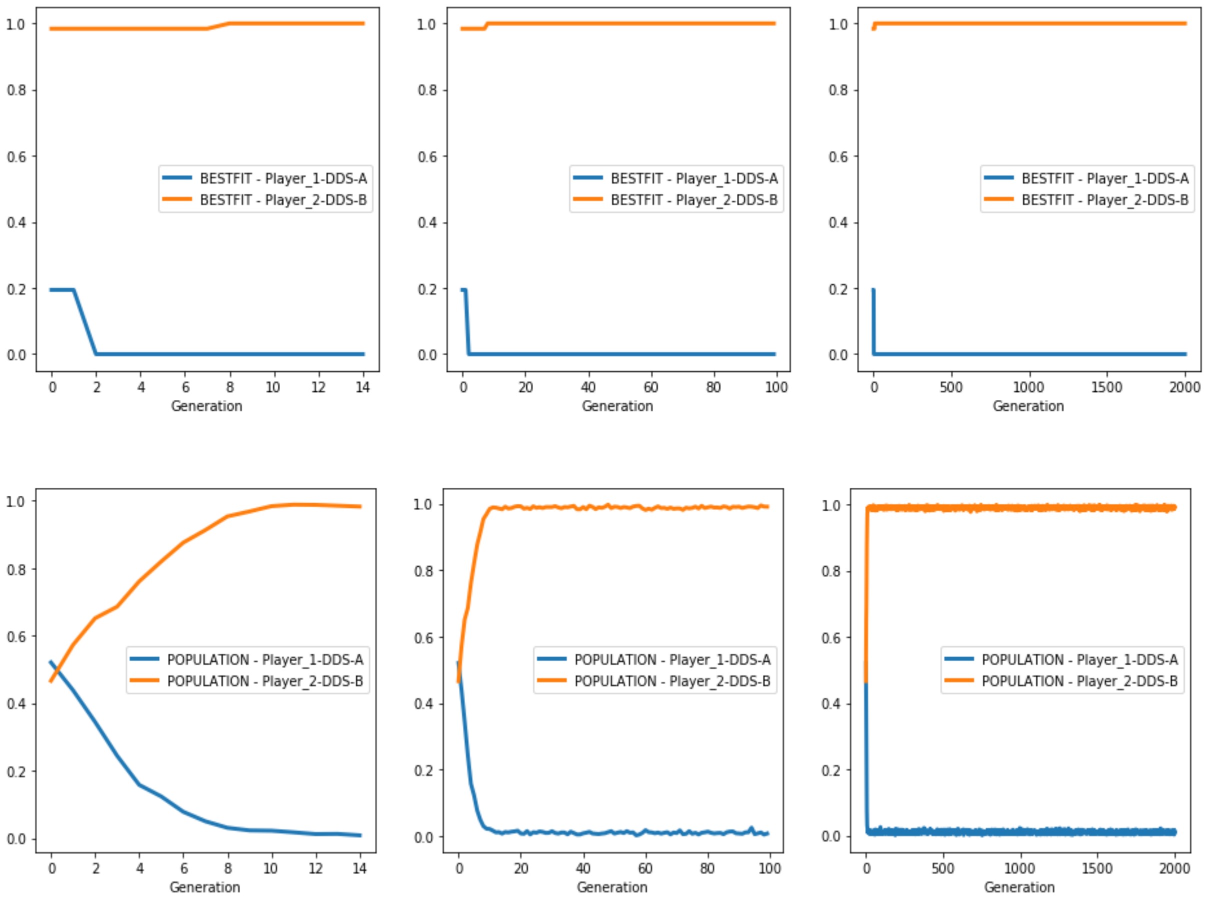


Figure 10: Convergence to an ESS of classic definition

Data obtained through the algorithm's run is presented in Figure 10. Each graph shows dynamic of evolution of an equilibrium strategies set for the model by demonstrating two lines: mixed strategy of the first player as a frequency of playing [A] move and mixed strategy of the second player as a frequency of playing [B] move. In the first row strategies of best fitted individuals in each generation are depicted, while in the second row average strategies of the subpopulations are shown. Graphs differ by column only in a number of generations plotted.

From the graph it could be readily seen that both best-fitted individuals' strategies and average subpopulations' strategies converge to one of the expected equilibrium $[B; B]$ – best-fitted individuals at 7th generation (never diverge in further generations) and average subpopulations' strategies at approximately 15th generation (fluctuate slightly around equilibrium point for all generations, never diverging significantly). It shows, that the algorithm is capable of convergence to an ESS of classic definition of (Smith, 1982).

8.4 Note on Convergence of Mixed Strategies Equilibria

8.4.1 Mixed Evolutionary Stable Strategy

ESS was defined originally for pure-strategist symmetrical game-theoretic models of coevolution in terms of replicator dynamic in differential equations (Smith, 1982), and actually stands for stable proportions of pure-strategist individuals with certain strategies in population, therefore could be treated as a stable state of population as a whole. Later, in the work of (Thomas, 1984) ESS concept was revisited; this work introduced stricter definition and presented analysis of actual individuals' strategies in the context of ESS; also, distinction between pure-strategist and mixed-strategist models was emphasized. Fundamental definition of an ESS suggests that it represents "an individual phenotype which can be established and maintained predominant by natural selection in evolutionary competition with alternative types" (Thomas, 1984) (Maynard Smith, et al., 1973), that presumed to be the case for both pure-strategist and mixed-strategist models. Pure ESS thus stands for monomorphic populations (populations, in which only one pure strategy is presented), and mixed ESS stands for polymorphic populations (populations, in which more than one pure strategy – regardless of being included in a mixed strategy or standing by itself – are presented). This means that as with the case of pure-strategist models mixed ESS in a mixed-strategist model suggests that stable set of individual phenotypes must be presented in the population.

One of the main findings on the topic is that there are could not exist mixed ESS in mixed-strategist models if mixed strategies payoff is a linear combination of payoffs of its components, unless mixed strategies sufficiently few in number (Thomas, 1985). Sufficiently few in this context means the space of mixed strategies available to individuals is of larger dimensionality than the space of possible pure strategies. This result (absence of mixed ESS) holds for virtually all mixed-strategist game-theoretic models. It is important because the algorithm uses mixed strategies by the design, hence treating all classic game-theoretic problems formulated as Normal Form Games and Games in Extensive Form as mixed-strategist models. On the other hand this result is a feature of ESS and relevant only for a particular class of game-theoretic models – linear models. Indeed, the algorithm that is proposed in this work is capable of working with non-linear models as well.

8.4.2 Mixed Evolutionary Stable Set as an Alternative to ES Strategy

As an extension of ESS concept, the concept of Evolutionary Stable Set was introduced in (Thomas, 1985). ES set concept defines stability for a population as a whole, characterized by a population strategy that is a weighted average of phenotypical content of the population, e.g. average of individuals' strategies. In the context of mixed-strategist models, each population strategy could be captured by arbitrarily large amount of population states. Population state is defined as a particular phenotypical profile of population, e.g. a set of individuals with particular strategies.

Stability of ES sets appears in the presence of a compact space of population states, from which, once converged, population would not diverge. In this case, population state of a population converged to ES set is expected to drift around ES set, preserving stable population strategy.

In the context of mixed-strategist models, it is useful to introduce the concept of degenerate model that is the same model treated as a pure-strategist one. It is shown that for a mixed-strategist model, every ESS of the degenerate variant of the model corresponds to an ES set of the mixed-strategist model (Thomas, 1985). Because of the definition of ESS, particularly its uniqueness in its own neighborhood, corresponding ES set would represent just a single population strategy. Therefore, ***our algorithm could capture a mixed ESS of a degenerate model through population dynamic***.

8.4.3 Uncorrelated Asymmetry

An uncorrelated asymmetry is a concept introduced in the context of evolutionary game theory for symmetrical games representing coevolutionary dynamic (Smith, 1982). It could be interpreted as follows: presence or absence of uncorrelated asymmetry determines whether agents in a game-theoretic model know which role do they play in it; or alternatively: whether agents know which role they have been assigned; or whether their strategies could be dependent on roles they have been assigned. It is important in symmetrical games. Due to (Selten, 1988) in a presence of information asymmetry (uncorrelated asymmetry) every ESS (if any exists) is a strict Nash Equilibrium (Samuelson, 1997 p. 49).

Our model by the design introduces uncorrelated asymmetry, because it establishes separate subpopulation for each decision-set in a given model, hence separate subpopulation for each player. Species evolving in separate subpopulations evolve independently from each other. Each fitness evaluation these species play the same roles. As it is stated in (Selten, 1988) “A sufficient condition for information asymmetry is satisfied if the two opponents in a contest always have roles which are different from each other”.

To tackle game-theoretic models in a setup with absence of uncorrelated asymmetry we developed a workaround extension for our algorithm. It should be noted that a scenario of an absence of uncorrelated asymmetry is established only for agents with identical decision-sets, which could naturally represent a case of agents being unaware of their role at fitness evaluation, because by the design there is no difference between them. Central idea of the extension is to make sure that subpopulations that corresponds to decision sets united by information symmetry are identical before fitness evaluations. That is each generation one of such subpopulations is selected and other subpopulations of the same information symmetry union are substituted with duplicates of the selected one. In this setup all decision-sets with information symmetry are represented by the same subpopulation. So because of random nature of the matching algorithm (Algorithm 1) and identical structure of the decision-sets of this subpopulation are evaluated as with random roles assigned, unless there are asymmetry in payoffs between decision-sets (this case is not considered in this paper and suggested for further research).

8.4.4 Empirical Results on Convergence to Mixed ESS in the Absence of Uncorrelated Asymmetry

To investigate a behavior of the algorithm on the task of locating Mixed ESS we use a classic for evolutionary game theory studies game called "Hawk Dove". It has two pure strategies Nash Equilibria $[Hawk;\ Dove]$ and $[Dove;\ Hawk]$ and one mixed Nash Equilibria $\left[\left(\frac{V}{C}\right) * Hawk + \left(1 - \frac{V}{C}\right) * Dove;\ \left(\frac{V}{C}\right) * Hawk + \left(1 - \frac{V}{C}\right) * Dove\right]$, when $V < C$; for a payoff matrix of the game please refer to Figure 11. In the presence of uncorrelated asymmetry, game has two ESS corresponding to pure NE. In the absence of uncorrelated asymmetry, game has only one ESS corresponding to mixed NE.

		Player 2	
		Hawk	Dove
Player 1	Hawk	(V-C)/2; (V-C)/2	V; 0
	Dove	0; V	V/2; V/2

Figure 11: HawkDove game's payoffs matrix

We construct two models based on Hawk Dove game with different payoff matrices, resulting in different proportions of $V \div C$ thus different Mixed ESS, to capture the difference in convergence of population strategy to Mixed ESS if there is any. HawkDoveSkewedHawk model in Figure 12 favors Hawk strategy in Mixed ESS and HawkDoveSkewedDove favors Dove strategy in Mixed ESS.

		Player 2	
		Hawk	Dove
Player 1	Hawk	-2.5; -2.5	15; 0
	Dove	0; 15	7.5; 7.5

Figure 12: HawkDoveSkewedHawk model's payoffs matrix

		Player 2	
		Hawk	Dove
Player 1	Hawk	-7.5; -7.5	5; 0
	Dove	0; 5	2.5; 2.5

Figure 13: HawkDoveSkewedDove model's payoffs matrix

We perform two runs of the algorithm on each model with and without uncorrelated asymmetry for 1000 generations. Then, for each run we construct a graph showing for each generation best fitted strategy in population and population strategy.

Algorithm run parameters:

popsize: 54, npairs: 20, ngames: 1, dropout_rate: 18, mutation_magnitude: 8

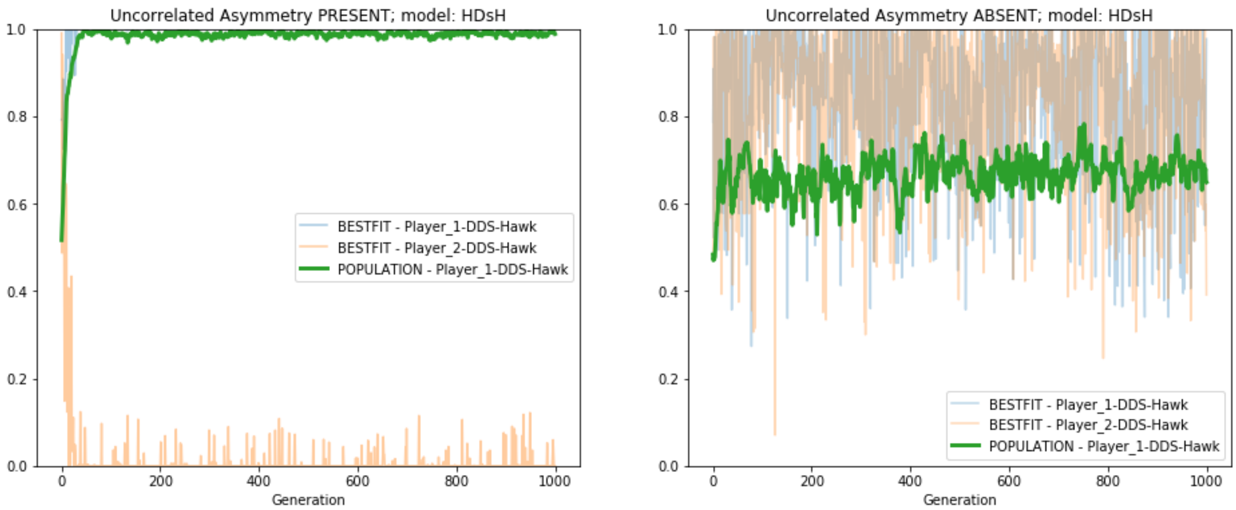


Figure 14: Mixed ESS in HawkDoveSkewedHawk

Algorithm run parameters:

popsize: 54, npairs: 20, ngames: 1, dropout_rate: 18, mutation_magnitude: 8

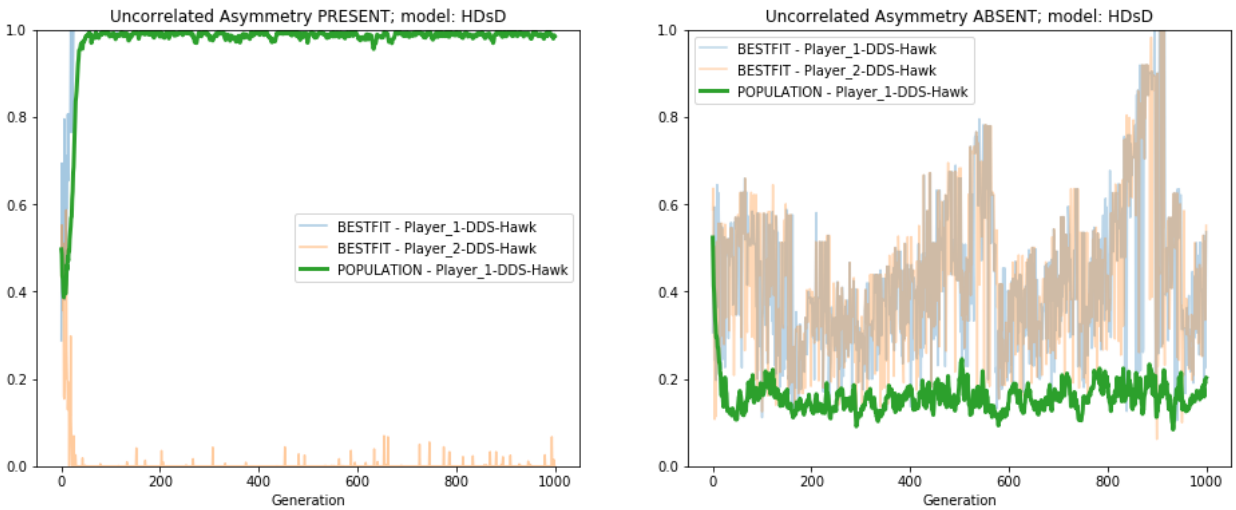


Figure 15: Mixed ESS in HawkDoveSkewedDove

We present the data obtained through running the algorithm in four graphs: two graphs on Figure 14 (for the model favoring Hawk strategy in mixed ESS) and two graphs Figure 15 (for the model favoring Hawk strategy in mixed ESS). Graph sets for both models are of identical structure. Graphs on the right-hand side show output of the algorithm running with modification simulating an absence of uncorrelated asymmetry. Graphs on the left-hand side show output of the algorithm running with default setup, thus introducing uncorrelated asymmetry to the model.

Each graph depicts the dynamic of evolution of an equilibrium strategies set for the model by three lines. The first one (light blue line) – mixed strategy of the best-fitted specie representing the first player as a frequency of playing $[Hawk]$ move. The second (light orange line) – mixed strategy of the best-fitted specie representing the second player as a frequency of playing $[Hawk]$ move. The third (bold green line) – average mixed population strategy as a frequency of playing $[Hawk]$ move.

As expected, under uncorrelated asymmetry both models converged in terms of individuals' strategies to one of the ESS represented by pure NE, in this case – $[Hawk;\ Dove]$. The same models evaluated in coevolutionary style with a setup of absence of uncorrelated asymmetry have shown non-convergence of individual strategies, which could be seen on graphs Figure 14 and Figure 15 by highly volatile random walk process of best-fitted strategy. In the same time on the same graphs it could be seen clearly, that population strategy represented on the graphs by frequency of $[Hawk]$ pure strategy quickly converges to low volatile random walk process around values approximately equal to theoretical Mixed ESS – 0.75 for HawkDoveSkewedHawk model and 0.25 for HawkDoveSkewedDove. It demonstrates the possibility to explore Mixed ESS by the proposed algorithm and in general, shows how ES sets could be explored by the same means.

8.5 Multiple Equilibria Statistics

We propose that our algorithm is capable of locating multiple equilibria of game-theoretic models. As it was discussed earlier, to do that we use Iterated Local

Search approach with random initial seeds. Theoretical foundation of such approach goes along with propositions of evolutionary game theory postulating, that players populations converge to ES sets if started in its neighborhood (Thomas, 1985). This justifies treatment of ILS with random seed approach as asymptotically capable of locating all equilibria of the model.

To utilize iterated local search method one should develop convergence metrics for the algorithm. Used as a termination criterion for the search process convergence metric could have a huge impact on total amount of computational resources consumed for the search process.

Such convergence metrics might differ from one application of the algorithm to another. The following factors should be considered while developing termination criteria for the algorithm: demanded level of accuracy in location of equilibria in terms of proportions of false positive and false negative returns and numerical precision of found equilibria position. It is advised to change several termination criteria between meta-heuristic ILS iterations, i.e. first roughly locate equilibria states, and then reinitialize algorithm in the located spots with different parameters and termination criteria to enhance numerical approximation of equilibria. It is also advisable to conduct supervised preliminary exploration of the problem and algorithm's performance on it to accumulate prior knowledge of the problem before initialization of the automated search procedure, because it might give insights regarding the problem's structure and guide parameters and termination criteria fine-tuning.

9 Conclusions and Further Research

In summary, this paper introduces a general-purpose numerical approach to investigate equilibria of mixed-strategist game-theoretic models in practical applications, capable of locating multiple equilibria, based on evolutionary game theory concepts and evolutionary optimization algorithms. One of the valuable features of the approach is that by the design it allows exploring the dynamic of evaluated models and making predictions of its convergence from a given state. The conceptual approach is complemented with an algorithmic implementation of it and tested with numerical computations. The influence of parameters of the algorithm on its performance is studied. The approach by the design is irrelevant to model specification and treat it as a black box. It is shown that the design of the algorithm allows it to tackle a broad range of models including linear, non-linear and noisy models with or without uncorrelated asymmetry and any combination of these. It is also shown and supported by empirical results that it is convergent to ES sets, including ES sets corresponding to ESS as defined by (Smith, 1982). In this context, the algorithm proved its use to analyze models on the scales of individual strategies and of population's dynamic as a whole.

The proposed here approach could be applied to analyze evolutionary equilibria and corresponding dynamic of a broad range of models involving social interactions in terms of agents' behavior, such as game-theoretic models, micro- and macroeconomic models with game-theoretic aspects and agent-based models.

To conclude, we suggest the following statements:

- Discrete numerical multi-population evolutionary optimization algorithm is capable of locating ES sets in mixed-strategist game-theoretic models
- Evolutionary approach to define equilibria of a social system could be applied to both linear and non-linear models with or without uncorrelated asymmetry and their combinations

- Evolutionary Stable Set is the most appropriate equilibria concept for numerical analysis of an arbitrary model, because it captures both agent- and system-level dynamics
- Numerical evolutionary algorithm is asymptotically expected to locate all equilibria of an arbitrary model through iterated local search process
- Running time of an automated assessment of an arbitrary model with an evolutionary based algorithm could be kept relatively low due to embarrassingly parallel nature of the algorithm

By far, it could be said that a proof of concept of the numerical evolutionary based approach to locating multiple equilibria of game-theoretic models is successfully implemented. The approach was developed with practical application in mind and designed capable of locating equilibria of an arbitrary black-box model, thus making its applied use straightforward.

At this stage, proposed algorithm might be suggested for use as a part of preliminary explorative analysis of an arbitrary model and is not intended to be used as the only tool for analysis of the model in production, because its properties and behavior are still not explored enough.

To overcome this and make it possible to use the algorithm for production analyzes of arbitrary models, we suggest that further research is needed to ensure reliable and predictable performance of the algorithm. One might want also to derive statistically reliable measures of its returns. For example, confidence intervals for numeric locations of found equilibria, estimated probability of found equilibria to be true equilibria of the model, estimated probability of presence unfound equilibria in the search space.

Finally, we suggest the following topics for further research:

- Extension of the algorithm to be able to handle decision-sets connected by an absence of uncorrelated asymmetry, but asymmetrical in payoffs structure

- Analytical investigation properties of possible attractors for the convergence dynamic of the algorithm
- Thorough comparative study of the algorithm’s performance on a broad range of model types with regard to its parameters
- Development a framework to predict the algorithm’s performance on an arbitrary model
- Development and assessment of performance of meta-heuristics for dynamical optimization of parameters of the algorithm dynamically during search runs
- Development and assessment of performance of meta-heuristics for the task of locating multiple equilibria utilizing derived knowledge of the problem during the search; guided local search framework is suggested
- Investigation of possible variations of the algorithm combining different heuristics of evolutionary optimization algorithms

10 References

Axelrod Robert The Evolution of Strategies in the Iterated Prisoner's Dilemma [Book Section] // Genetic Algorithms and Simulated Annealing / book auth. Davis Lawrence. - Los Altos, CA : Morgan Kaufman, 1987.

B. Morsky C.T. Bauch Truncation selection and payoff distributions applied to the replicator equation [Journal] // Journal of Theoretical Biology. - 2016. - pp. 383-390.

Bull L. On coevolutionary genetic algorithms [Journal] // Soft Computing. - 2001. - pp. 201-207.

Cantú-Paz Erick Efficient and Accurate Parallel Genetic Algorithms [Book]. - [s.l.] : Springer US, 2001.

Daniel Kahneman Thomas Gilovich Heuristics and Biases : The Psychology of Intuitive Judgment [Book]. - New York, NY, USA : Cambridge University Press, 2002.

Fogarty Lawrence BullTerence C. Co-Evolving Communicating Classifier Systems for Tracking [Book Section] // Artificial Neural Nets and Genetic Algorithms / book auth. Albrecht R.F. Reeves C.R., Steele N.C.. - Vienna : Springer, 1993.

Husbands Phil Distributed Coevolutionary Genetic Algorithms for Multi-Criteria and Multi-Constraint Optimisation [Book Section] // Evolutionary Computing. AISB EC 1994. Lecture Notes in Computer Science, vol 865 / book auth. T.C. Fogarty. - Berlin, Heidelberg : Springer, 1994.

I.A.Ismail N.A.El_Ramly , M.M.El_Kafrawy , M.M.Nasef Game Theory Using Genetic Algorithms [Conference] // Proceedings of the World Congress on Engineering 2007, WCE 2007. - London, U.K. : [s.n.], 2007.

Iris Lorscheid Matthias Meyer Agent-based mechanism design – investigating bounded rationality concepts in a budgeting context [Journal] // Team Performance Management. - 2017. - pp. 13-27.

Jacek B. Krawczyk Stanislav Uryasev Relaxation algorithms to find Nash equilibria with economic applications [Journal] // Environmental Modeling and Assessment. - 2000. - pp. 63-73.

Jian Chi Yanfei Liu Multi-objective Genetic Algorithm based on Game Theory and its Application [Conference] // 2nd International Conference on Electronic & Mechanical Engineering and Information Technology (EMEIT-2012). - Paris, France : Atlantis Press, 2012. - pp. 2341-2344.

John von Neumann Oskar Morgenstern Theory of Games and Economic Behavior [Book]. - [s.l.] : Princeton University Press, 1944.

Jong Kenneth A. De Evolutionary Computation [Book]. - Cambridge, Massachusetts; London, England : The MIT Press, 2006.

Ken'ichiro Tanakaa Alexis Akira Toda Discrete approximations of continuous distributions by maximum entropy [Journal] // Economics Letters. - 2012. - pp. 445-450.

Keynes John Maynard The General Theory of Employment, Interest and Money [Book]. - [s.l.] : Palgrave Macmillan, 1936.

Kwee-Bo Sim Ji-Yoon Kim Solution of multiobjective optimization problems: coevolutionary algorithm based on evolutionary game theory [Journal] // Artificial Life and Robotics. - 2004. - pp. 174-185.

Lucas Robert Econometric Policy Evaluation: A Critique [Book Section] // The Phillips Curve and Labor Markets / book auth. Brunner K. and Meltzer A. - New York : American Elsevier, 1976.

M. Sefrioui J. Periaux Nash Genetic Algorithms : examples and applications [Conference] // Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512). - La Jolla, CA, USA : IEEE, 2000. - pp. 509-516.

Marks Robert E. Playing Games with Genetic Algorithms [Article] // Evolutionary Computation in Economics and Finance. - 2002. - pp. 31-44.

Marshall Alfred Principles of Economics [Book]. - London, UK : Macmillan, 1890.

Mattheos K. Protopapas Francesco Battaglia, Elias B. Kosmatopoulos Coevolutionary Genetic Algorithms for Establishing Nash Equilibrium in Symmetric Cournot Games [Journal] // Advances in Decision Sciences. - 2010.

Maynard Smith J. and Price G.R. The logic of animal conflict [Journal] // Nature. - 1973. - pp. 15-18.

Milos S. Stankovic Karl H. Johansson, and Dusan M. Stipanovic Distributed Seeking of Nash Equilibria With [Journal] // IEEE TRANSACTIONS ON AUTOMATIC CONTROL. - 2012. - pp. 904-919.

Morrison Clarence C. Cournot, Bertrand, and modern game theory [Journal] // Atlantic Economic Journal. - 1998. - pp. 172-174.

Moshe Hoffman Sigrid Suetens, Uri Gneezy, Martin A. Nowak An experimental investigation of evolutionary dynamics in the Rock-Paper-Scissors game [Journal] // Scientific Reports. - 2005.

Murphy Antoin E. John Law and Richard Cantillon on the circular flow of income [Journal] // Journal of the History of Economic Thought. - 1993. - pp. 47-62.

Nash John Non-Cooperative Games [Journal] // The Annals of Mathematics. - 1951. - pp. 286-296.

Neely Michael J. A Lyapunov Optimization Approach to Repeated [Conference] // Proc. Allerton Conference on Communication, Control, and Computing. - 2013.

Nelson Richard R. Behavior and cognition of economic actors in evolutionary economics [Journal] // Journal of Evolutionary Economics. - [s.l.] : Springer-Verlag Berlin Heidelberg, 2015.

Noam Nisan Tim Roughgarden, Eva Tardos, Vijay V. Vazirani Algorithmic Game Theory [Book]. - New York, USA : Cambridge University Press, 2007.

Olivier Bournez Johanne Cohen Learning Equilibria in Games by Stochastic Distributed Algorithms [Book Section] // Computer and Information Sciences III / book auth. E. Gelenbe R. Lent. - London, UK : Springer-Verlag London, 2013.

Paul Frihauf Miroslav Krstic, Tamer Basar Nash Equilibrium Seeking in Noncooperative Games [Journal] // IEEE TRANSACTIONS ON AUTOMATIC CONTROL. - [s.l.] : IEEE, May 2012. - 5 : Vol. 57. - pp. 1192-1207.

Peters Hans Game Theory, A Multi-Leveled Approach [Book]. - Berlin, Germany : Springer-Verlag Berlin Heidelberg, 2015.

Preuss Mike Multimodal Optimization by Means of Evolutionary Algorithms [Book]. - Münster, Germany : Springer International Publishing Switzerland, 2015.

Richard D McKelvey Andrew McLennan Computation of Equilibria in Finite Games [Book Section] // Handbook of Computational Economics / book auth. Karl Schmedders Kenneth L. Judd. - [s.l.] : Elsevier, 1996.

Rizvi S. Abu Turab The Sonnenschein-Mantel-Debreu [Journal] // History of Political Economy. - 2006. - pp. 228-245.

Rodica Lung D. Dumitrescu An Evolutionary Model for Solving Multiplayer Noncooperative Games [Conference] // Knowledge Engineering: Principles And Techniques; Proceedings of the International Conference on Knowledge

Engineering, Principles and Techniques, KEPT2007. - Cluj-Napoca (Romania) : [s.n.], 2007. - pp. 209–216.

Roughgarden Tim Computing equilibria: a computational complexity perspective [Journal] // Economic Theory. - January 2010. - pp. 193-236.

Samuelson Larry Evolutionary Games and Equilibrium Selection [Book]. - Cambridge, MA : The MIT Press, 1997.

Selten Reinhard A Note on Evolutionarily Stable Strategies in Asymmetric Animal Conflicts [Book Section] // Models of Strategic Rationality. - [s.l.] : Springer Netherlands, 1988.

Sevan G. Ficici Ofer Melnik, and Jordan B. Pollack A Game-Theoretic and Dynamical-Systems Analysis of Selection Methods in Coevolution [Journal] // IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION. - 2005. - pp. 580-601.

Smith Adam The Wealth of Nations [Book]. - London, UK : W. Strahan and T. Cadell, 1776.

Smith John Maynard Evolution and the Theory of Games [Book]. - [s.l.] : Cambridge University Press, 1982.

Srihari Govindan Robert Wilson Computing Nash equilibria by iterated polymatrix approximation [Journal] // Journal of Economic Dynamics & Control. - 2004. - pp. 1229-1241.

Stengel Bernhard von Computation of Nash equilibria in finite games: introduction to the symposium [Journal] // Economic Theory. - 2010.

Suheyla Ozyildirim Nedim M. Alemdar Learning the optimum as a Nash equilibrium [Journal] // Journal of Economic Dynamics & Control. - 2000. - pp. 483-499.

Thomas Bernhard Evolutionarily Stable Sets in Mixed-Strategist Models [Journal] // Theoretical Population Biology. - 1985. - pp. 332-341.

Thomas Bernhard Evolutionary stability: States and strategies [Journal] // Theoretical Population Biology. - 1984. - pp. 49-67.

Thomas Bernhard On evolutionarily stable sets [Journal] // Journal of Mathematical Biology. - 1985. - pp. 105-115.

Veisi Nona Helmi and Gelareh A Multi-Modal Coevolutionary Algorithm for Finding All Nash Equilibria [Journal] // Ubiquitous Information Technologies and Applications. - 2012. - pp. 21-29.

Wei-Kai Lin Tian-Li Yu The Co-Evolvability of Games in Coevolutionary Genetic Algorithms [Report]. - Taiwan : Taiwan Evolutionary Intelligence Laboratory (TEIL), 2009.