2013

Computation and analysis of evolutionary game dynamics

Yiping Hao
Iowa State University

Follow this and additional works at: http://lib.dr.iastate.edu/etd

Part of the Applied Mathematics Commons

Recommended Citation
Hao, Yiping, "Computation and analysis of evolutionary game dynamics" (2013). Graduate Theses and Dissertations. 13356.
http://lib.dr.iastate.edu/etd/13356

This Dissertation is brought to you for free and open access by the Graduate College at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Computation and analysis of evolutionary game dynamics

by

Yiping Hao

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Applied Mathematics

Program of Study Committee:
Zhijun Wu, Major Professor
Stephen Willson
Alexander Roitershtein
David Fernández-Baca
Stanley Harpole

Iowa State University
Ames, Iowa
2013
Copyright © Yiping Hao, 2013. All rights reserved.
DEDICATION

I would like to dedicate this dissertation to my parents, Wensi Hao and Aihua Wang, for their encouragement and support while I was far away from home during my Ph.D. program.

I would also like to dedicate it to my wife Jing Wang without whose love and support I would not have been able to complete this work.
TABLE OF CONTENTS

LIST OF TABLES ................................................................. v
LIST OF FIGURES .............................................................. vi
ACKNOWLEDGEMENTS ....................................................... vii
ABSTRACT ................................................................. viii

CHAPTER 1. INTRODUCTION ............................................... 1

CHAPTER 2. GAME THEORETICAL MODELING OF EVOLUTIONARY DYNAMICS ........................................ 4
  2.1 Introduction ............................................................. 4
  2.2 Nash equilibrium ....................................................... 4
  2.3 Necessary and sufficient conditions of Nash equilibrium for evolutionary games 9

CHAPTER 3. INVERSE GAME BASED ON REPLICATOR DYNAMICS 19
  3.1 Introduction ............................................................. 19
  3.2 Data transformation ................................................... 19
  3.3 Smoothing and least squares ....................................... 22
  3.4 Inferences based on parametric bootstraps ....................... 23

CHAPTER 4. DIRECT EVOLUTIONARY GAMES ......................... 25
  4.1 Introduction ............................................................. 25
  4.2 Specialized Snow-Shapley algorithm ............................. 25
  4.3 Specialized Lemke-Howson algorithm ............................ 29
  4.4 Complementarity method ............................................ 34
CHAPTER 5. EVOLUTIONARY GAME DYNAMICS .......................... 37
  5.1 Introduction .......................................................... 37
  5.2 Replicator dynamics ................................................. 37
  5.3 Stability conditions: Nash equilibrium, KKT point, and ESS ................. 38

CHAPTER 6. SPECIAL TOPIC: SPARSE AND DENSE NASH EQUILIBRIA .................. 46
  6.1 Introduction .......................................................... 46
  6.2 Dual method for direct evolutionary games ........................... 47

CHAPTER 7. NUMERICAL EXAMPLES ................................. 51
  7.1 Gene mutations for malaria resistance ........................... 51
  7.2 An invading model in vegetation dynamics ........................ 55

CHAPTER 8. SOLVER IMPLEMENTATION: TEDA ...................... 58
  8.1 Toolbox for evolutionary dynamics analysis (TEDA) .................... 58
  8.2 Example of inverse evolutionary game ................................ 59
  8.3 Example of direct evolutionary game ................................ 63
  8.4 Stability analysis .................................................... 68
  8.5 Dynamics simulation ................................................ 69

CHAPTER 9. SUMMARY ................................................. 71

BIBLIOGRAPHY ............................................................ 73
LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 6.1</td>
<td>Computation time (in seconds) for direct game by specialized Snow-</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Shapley method</td>
<td></td>
</tr>
<tr>
<td>Table 6.2</td>
<td>Computation time (in seconds) for direct game by dual method</td>
<td>50</td>
</tr>
<tr>
<td>Table 7.1</td>
<td>Payoff matrix from [47] regarding plant successions problem</td>
<td>57</td>
</tr>
<tr>
<td>Table 8.1</td>
<td>Data from [48] regarding Barnes’ problem</td>
<td>60</td>
</tr>
<tr>
<td>Table 8.2</td>
<td>Payoff matrix from [47] regarding plant successions problem</td>
<td>63</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

| Figure 8.1 | Main interface of TEDA.  | 59 |
| Figure 8.2 | Parameter estimation for Barnes’ Problem | 60 |
| Figure 8.3 | Margin distribution of estimated components of payoff matrix for the Barnes’ problem using evolutionary game modeling. | 61 |
| Figure 8.4 | Inference statistics of the estimated payoff matrix | 62 |
| Figure 8.5 | 95% credible bands for estimations of the Barnes’ Problem | 62 |
| Figure 8.6 | Payoff matrix inputting in TEDA | 64 |
| Figure 8.7 | Method selection for solving direct evolutionary games in TEDA | 64 |
| Figure 8.8 | Solving a direct game by using complementarity method | 65 |
| Figure 8.9 | Solving a direct game with sparsity options | 65 |
| Figure 8.10 | Nash equilibria found by complementarity method | 66 |
| Figure 8.11 | Nash equilibria found by Snow-Shapley method | 67 |
| Figure 8.12 | Nash equilibria found by Lemke-Howson method | 68 |
| Figure 8.13 | Stability checking for equilibria | 69 |
| Figure 8.14 | Dynamics simulation based on replicator equations system | 70 |
ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere thanks and gratitude to my major professor, Dr. Zhijun Wu, for his guidance and endless support throughout this research and the writing of this thesis. His insights and words of encouragement have often inspired me and renewed my hopes for completing my graduate education. I would like to thank my committee members: Dr. David Fernández-Baca, Dr. Stan Harpole, Dr. Alexander Roitershtein, and Dr. Stephen Willson, for their efforts and contributions to this work. Especially, I would also like to thank Dr. Wen Zhou, for his collaboration and valuable contributions to this work. I would additionally like to thank the Department of Mathematics at Iowa State University for providing me the financial support throughout my PhD study. All my friends, colleagues and research group members, who have ever assisted me, are gratefully appreciated as well.
ABSTRACT

Biological processes are usually defined based on the principles of replication, mutation, competition, adaption, and evolution. In evolutionary game theory, such a process is modeled as a so-called evolutionary game, which not only provides an alternative interpretation of dynamical equilibrium in terms of the game nature of the process, but also bridges the stability of the biological process with the Nash equilibrium of the evolutionary game. Computationally, the evolutionary game models are described in terms of inverse and direct games, which are estimating the payoff matrix from data and computing the Nash equilibrium of a given payoff matrix respectively. We discuss the necessary and sufficient conditions for the Nash equilibrium states, and derive the methods for both inverse and direct games in this thesis. The inverse game is solved by a non-parametric smoothing method and penalized least squares method, while different schemes for the computation of the direct game are proposed including a specialized Snow-Shapley algorithm, a specialized Lemke-Howson algorithm, and an algorithm based on the solution of a complementarity problem on a simplex. Computation for the sparsest and densest Nash equilibria is investigated. We develop a new algorithm called dual method with better performance than the traditional Snow-Shapley method on the sparse and dense Nash equilibrium searching. Computational results are presented based on examples. The package incorporating all the schemes, the Toolbox of Evolution Dynamics Analysis (TEDA), is described.
CHAPTER 1. INTRODUCTION

Biological species or microorganisms such as viruses, bacteria, plants, or animals, replicate, mutate, compete, adapt, and evolve, which form the building blocks of evolutionary dynamics of life. To quantitatively understand this sophisticated process, different theoretical models have been developed, and they reveal significant biological insights [8; 11; 13; 20; 21; 23; 24; 26; 27; 38; 39; 43; 49; 52; 53; 54]. Modeling biological interaction and evolution process via game theoretical approaches have drawn a great amount of attention during the past three decades [1; 2; 9; 14; 15; 16; 22; 25; 31; 33; 34; 30; 46; 51]. Based on the five principles of replication, mutation, competition, adaption, and evolution, biological process is modeled by a so-called evolutionary game in the context of evolutionary game theory. The evolutionary game is a special case of symmetric games in which the strategies adapted by the players are identical and a common payoff matrix is defined [16; 51].

Game theoretical modeling of an evolutionary process provides an alternative way to interpret the dynamical equilibrium by adapting the concepts of Nash equilibrium from game theory [16]. It bridges the game theoretical nature of the biological process with the stabilities of biological dynamical systems defined by the payoff matrix of a certain game, which are known as the replicator equations [16; 30]. On the other hand, evolutionary game modeling brings computational challenges, among which two key questions are, first of all, how to determine the payoff matrix for dynamical data given in the form of a time series, which is the problem of the inverse evolutionary game, and more important, how to find the Nash equilibrium for the evolutionary game defined by the known payoff matrix, which is the direct evolutionary game.

In this thesis, I first look at the optimality conditions for evolutionary games in chapter
2. Consider an evolutionary game defined by a symmetric fitness matrix $A \in \mathbb{R}^{n \times n}$. Let $S_n = \{x \in \mathbb{R}^n : x \geq 0, \sum x_i = 1\}$ be the set of all mixed strategies. Solving a direct evolutionary game is to find an optimal strategy $x^* \in S_n$ called a Nash equilibrium strategy such that

$$x^{*T}Ax^* \geq x^TAx^*, \quad \text{for all } x \in S_n. \quad (1.1)$$

We call this problem a symmetric evolutionary game. This problem has important applications in population genetics, where it can be used to model and study distributions of genes in given populations when they are under certain selection pressures. We explore the necessary and sufficient conditions for the equilibrium states in detail. These conditions are applied to solving direct evolutionary games later in chapter 4.

After that, in order to obtain the fitness matrix for solving direct games, I investigate the inverse game in chapter 3. An inverse game targets on recovering the payoff matrix $A$ for the evolutionary game model based on the data and replicator equations, whose parameters are the components of the payoff matrix. To obtain the estimation and inference on $A$, problems of data manipulation and data smoothing are solved using the least squares method and parametric bootstraps sampling techniques.

In chapter 4, I discuss computational schemes for solving direct games, including a specialized Snow-Shapley algorithm [40], a specialized Lemke-Howson algorithm [18], and a searching algorithm based on the solution of a complementarity problem on a simplex. The Snow-Shapley procedure [40] is a classical algorithm for finding all extreme optimal strategies via exhausting all subsystems and is a purely combinatorial algorithm. The Lemke-Howson algorithm [18] is a classical simplex type algorithm developed to search for Nash equilibria of two-player, finite-strategy games. For the evolutionary game, we make some assumptions to obtain the specialized Snow-Shapley and Lemke-Howson methods. Using results from chapter 2, the necessary and sufficient conditions of Nash equilibrium provide complementarity conditions that lead to the complementarity method. It solves the direct evolutionary game by solving a reformulated quadratic programming problem.

After addressing the direct and inverse game, I introduce the evolutionary game dynamics.
The dynamics of the replicator equation system are investigated in detail. The relationships among Nash equilibria, KKT points, and ESS are discussed. Chapter 5 also provides another way for approaching to the Nash equilibrium by utilizing the dynamics of the replicator equation system.

In chapter 6, I focus on the special topic: the sparsest and densest Nash equilibria in direct evolutionary games. Based on the necessary and sufficient conditions of Nash equilibrium discussed in chapter 2, we derive a new algorithm, dual method, with the same completeness as the specialized Snow-Shapley method on Nash equilibria searching but with better performance.

Numerical examples including a gene mutation for malaria resistance and a plant succession problem are presented in chapter 7. Both of these two are direct evolutionary games. The different schemes for solving the direct evolutionary games discussed in chapter 4 are applied to these examples. Finally, the solver implementation is presented in chapter 8. The toolbox for evolutionary dynamics analysis (TEDA) is shown with functions and modules, applications in inverse and direct game, stability analysis for equilibria, and dynamics analysis.

In summary, I present our work on the evolutionary game modeling of biological processes and theoretical results about the necessary and sufficient conditions of Nash equilibrium in chapter 2. In chapter 3, our work on solving inverse games is presented. In Chapter 4, I introduce our work on solving direct games with specialized Snow-Shapley, specialized Lemke-Howson, and a searching algorithm based on the solution of a complementarity problem on a simplex. Chapter 5 is devoted to evolutionary game dynamics. In chapter 6, the special topic, searching the sparsest and densest equilibria in evolutionary games is discussed. Finally, numerical examples and the implementation of a Matlab toolbox are demonstrated in chapters 7 and 8.
CHAPTER 2. GAME THEORETICAL MODELING OF EVOLUTIONARY DYNAMICS

2.1 Introduction

Established by evolutionary biologists including John Maynard Smith and co-workers [26], the population-based evolutionary game theory has found many applications in biological and non-biological fields like economics and learning theory, and presents an important enrichment of classical game theory. The classical game theory is centered on the concept of rational decision making. It deals with a rational individual, who is engaged in a given interaction or game with other players and has to decide between different options (strategies), in order to maximize a payoff that depends on the strategies of the co-players. The evolutionary game theory deals with entire populations of players, all programmed to use some strategy or type of behavior. Strategies with high payoff will spread within the population, which can be achieved by learning, by copying or inheriting strategies, or even by infection. The payoffs depend on the actions of the co-players and hence on the frequencies of the strategies within the population. Since these frequencies change according to the payoffs, this yields a feedback loop. The dynamics of this feedback loop is the object of evolutionary game theory. The evolutionary game is a game-theoretical model to study the evolutionary dynamics of biological or social systems. It models the population interaction evolving along time by treating the population frequency as strategy to which some payoff matrices are assigned.

2.2 Nash equilibrium

We start by looking at the game with only two players, I and II, each with a finite set of options or pure strategies, set $N$ and $M$ for player I and II, respectively. Denote by $a_{ij}$ and
The payoff for player I and II when player I uses strategy $i \in N$ and II uses $j \in M$. Thus the payoffs are given by the $n \times m$-matrices $A$ and $B$, with $n$ and $m$ as the cardinalities of the sets of pure strategies.

The mixed strategy of player I that consists of using $i \in N$ with probability $x_i$ is denoted by the vector $x = (x_1, \ldots, x_n)^T$, which is an element of the unit simplex $S_n$ spanned by the vectors $e_i$ of the standard unit base: these vectors will be identified with the elements of $N$. Similarly, the unit simplex $S_m$ spanned by the vectors $f_j$ corresponds to the set of mixed strategies for player II. If player I uses $x \in S_n$ and II uses $y \in S_m$, then the former has the expected payoff $x^T Ay$ and the latter $x^T By$. The strategy $x \in S_n$ is said to be a best response to $y \in S_m$ if
\[
z^T Ay \leq x^T Ay, \quad \text{for all} \quad z \in S_n.
\] (2.1)
The set of all best replies to $y$ is denoted by $BR(y)$. A pair $(x, y) \in S_n \times S_m$ is a Nash equilibrium if $x \in BR(y)$ and $y \in BR(x)$. As we shall see, a simple fixed point argument shows that such a Nash equilibrium always exists. The pair is said to be a strict Nash equilibrium if $x$ is the unique best reply to $y$ and vice versa. Necessarily, strict Nash equilibria are of the form $(e_i, f_j)$. If two strategies form a Nash equilibrium, none of the players has an incentive to deviate unilaterally. In this sense, such an outcome satisfies a consistency condition.

In order to transfer this to a population setting, it is convenient to consider the case where the two players are interchangeable individuals within the population, i.e. to consider only the case where the two players do not appear in different roles such as buyer and seller, but have the same strategy set and the same payoff matrix. More precisely, we first consider symmetric games, defined by $N = M$ and $A = B^T$. For symmetric games, players cannot be distinguished and only symmetric pairs $(x, x)$ of strategies are of interest. We shall therefore say that strategy $x \in S_n$ is a Nash equilibrium if
\[
z^T Ax \leq x^T Ax, \quad \text{for all} \quad z \in S_n,
\] (2.2)
i.e. if $x$ is a best reply to itself. The equilibrium is said to be strict if equality holds only for $z = x$. 
Denote \( x \in S_n = \{ x \in \mathbb{R}^n : \sum_{i=1}^{n} x_i = 1, x_i \geq 0, i = 1, \ldots, n \} \) to be the population vector of all species in a closed social or ecosystem. Biologically, 

\[
\frac{1}{x_i} \frac{dx_i}{dt}, \quad i = 1, \ldots, n, \tag{2.3}
\]

defines the vector of fitness of all species. In the context of evolutionary game theory [16], it is conventional that 

\[
\frac{1}{x_i} \frac{dx_i}{dt} = f_i(x, A(x)) - \phi \tag{2.4}
\]

where \( f(x, A(x)) \) represents the vector of fitness as a function of population frequencies and payoff matrix \( A(x) \), and \( \phi \) denotes the average effective fitness for the purpose of balancing the system, which is usually modeled by 

\[
\phi := \int f(x)dx = \sum_{i=1}^{n} f_i(x_i)x_i. \tag{2.5}
\]

The population frequency \( x \) in fact defines a probability measure on \( S_n \). In general 

\[
f(x, A(x)) := \int_{S_n} A(x)dx, \tag{2.6}
\]

which is well-defined provided \( A(x) \in L^1(S_n) \). More specifically, for the evolutionary game with finite number of species and constant payoff matrix along time 

\[
f(x, A) = Ax \tag{2.7}
\]

and the average payoff is written as 

\[
\phi = x^TAx = \sum_{i,j=1}^{n} A_{ij}x_i x_j. \tag{2.8}
\]

The evolutionary game modeling for a finite number of species/strategies evolving in time with constant payoff matrix is therefore summarized as the well-known system of replicator equations:

\[
\frac{dx_i}{dt} = x_i((Ax)_i - x^TAx), \quad x_i \geq 0, \quad \sum_{i} x_i = 1 \tag{2.9}
\]

In the context of dynamical system theory, the distribution of species/strategies \( x \) is a fixed point of the system given

\[
x_i((Ax)_i - x^TAx) = 0, \quad \forall i, \tag{2.10}
\]
and the system is claimed to reach an equilibrium.

In the context of game theory, a strategy \( x \in S_n \) is called an optimal strategy if

\[
    x^T A x \geq y^T A x, \quad \text{for all } y \in S_n,
\]

and we say the population reaches a Nash equilibrium \([16; 51]\). \((2.10)\) and \((2.11)\) establish the connections between the replicator equation system and the symmetric evolutionary game by means of the equilibrium and are shown in \([16]\) that

1. \( x \) is the fixed point of \((2.9)\) if \( x \in S_n \) is a Nash equilibrium of the game defined by the payoff matrix \( A \)

2. \( x \) is a Nash equilibrium of the game defined by the payoff matrix \( A \) if \( x \) is Lyapunov stable equilibrium of \((2.9)\) or \( x \) is the \( \omega-\)limit of an orbit \( x(t) \in \text{int}(S_n) \).

The above results can be further elaborated by means of the following inclusion

\[
    S \subset N \subset F
\]

where \( S \) and \( F \) are the sets of Lyapunov stable equilibriums and fixed points for the replicator equation system \((2.9)\), and \( N \) denotes the set of Nash equilibriums of evolutionary game defined by the same payoff matrix \( A \) in \((2.9)\). It is well known that the Nash equilibrium exists for the games of all mixed and pure strategies with constant payoff matrix \( A \) \([28]\), so that \( F \neq \emptyset \) and \( x \in N \) if \( x \in S \). Therefore, the inclusion \((2.12)\) does not only imply the connection between the symmetric evolutionary game and replicator equation system theoretically, but also provide a tool to check whether or not a fixed point of \((2.9)\) is also a Nash equilibrium. We have utilized it in a Matlab toolbox that will be shown in chapter 6. More details about the connection of the symmetric evolutionary game, the replicator equation system, and also the generalized knapsack problem will be presented in section 2.2 and 2.3.

If the fitness matrix of a symmetric game itself is symmetric, the game is called a doubly symmetric game \([51]\). An evolutionary game with a symmetric fitness matrix is a doubly symmetric game, which is what we call a symmetric evolutionary game. In any case, it is well known that a mixed strategy \( x^* \) for an evolutionary game is optimal if and only if \( \pi(x^*, x^*) = \)
A mixed strategy $\mathbf{x}^* \in S_n$ is optimal for an evolutionary game if and only if for some scalar $\lambda^*$,

1. $x_i^* \geq 0, \quad \lambda^* - \pi(e_i, \mathbf{x}^*) \geq 0$

2. $x_i^*(\lambda^* - \pi(e_i, \mathbf{x}^*)) = 0$ for $i = 1, 2, \cdots, n$.

An important concept in evolutionary game theory is the evolutionary stability of an optimal strategy. It characterizes the ability of a population to resist small changes or invasions when at equilibrium. Let $\mathbf{x}^*$ be the optimal strategy. Then, the population is in the equilibrium state $\mathbf{x}^*$. Let $\mathbf{x}'$ be another arbitrary strategy. Mix $\mathbf{x}^*$ and $\mathbf{x}' \neq \mathbf{x}^*$ so that the population changes to a new state, $\epsilon \mathbf{x}' + (1-\epsilon)\mathbf{x}^*$, for some small fraction $\epsilon > 0$. Then, $\mathbf{x}^*$ is said to be evolutionarily stable if it remains as a better respond to the new ‘invaded’ population state. More accurately, we have the following definition.

**Definition 2.2.2.** An optimal strategy $\mathbf{x}^* \in S_n$ for an evolutionary game is evolutionarily stable if there is a small number $\epsilon' \in (0, 1)$ such that for any $\mathbf{x}' \in S_n, \mathbf{x}' \neq \mathbf{x}^*$,

$$
\pi(\mathbf{x}^*, \epsilon \mathbf{x}' + (1-\epsilon)\mathbf{x}^*) > \pi(\mathbf{x}, \epsilon \mathbf{x}' + (1-\epsilon)\mathbf{x}^*), \quad \text{for all } \mathbf{x} \in S_n \text{ and } 0 < \epsilon \leq \epsilon'. \quad (2.13)
$$

The equilibrium of an evolutionary game can also be viewed as the result of a dynamic process of the population change, where the game can be started from any population state and played infinitely many times until the population reaches equilibrium. Let $\mathbf{x}$ be a population state at time $t$, with $x_i$ being the portion of species $i$ in the population. Assume that $\mathbf{x}$ is a continuous function of $t$. Then the change of $\mathbf{x}$ can be described by a so-called system of replicator equations, based on the principle that the rate of replication of species $i$ is proportional to the fitness gain of species $i$ relative to the average fitness of the population [16]. Let $A$ be the fitness matrix. Then, the fitness of species $i$ in population $\mathbf{x}$ would be $\pi(e_i, \mathbf{x}) = (Ax)_i$, which is the average of the fitness values in the $i$-th row of $A$.
weighted by their frequencies given in \( x \), while the average fitness of all species types would be \( \sum_i x_i \pi(e_i, x) = \sum_i x_i (Ax)_i = x^T Ax = \pi(x, x) \), which is the average of the fitness values of all species \( i = 1, \ldots, n \) weighted again by their corresponding frequencies given in \( x \). Therefore, a system of replicator equations can be defined as

\[
\frac{dx_i}{dt} = x_i (\pi(e_i, x) - \pi(x, x)), \quad x_i \geq 0, \quad \sum_i x_i = 1, \quad i = 1, \ldots, n. \tag{2.14}
\]

Note that an evolutionary game (2.11) always corresponds to a system of replicator equations (2.14) and vice versa. It is well known that an optimal strategy for (2.11) must be a fixed point of the corresponding system of replicator equations (2.14). The converse may not necessarily be true. However, an asymptotically stable equilibrium point of (2.14) must be an optimal strategy for (2.11). Again, the converse may not necessarily be true. Also, an evolutionarily stable strategy of (2.11) must be an asymptotically stable equilibrium point of (2.14), although the converse may not always be true.

**Theorem 2.2.3.** [50] Let an evolutionary game be given in (2.11) and a corresponding system of replicator equations in (2.14). Let \( O \) and \( E \) be the sets of optimal strategies and evolutionarily stable strategies for (2.11), respectively. Let \( F \) and \( A \) be the sets of fixed points and asymptotically stable equilibrium points of (2.14), respectively. Then,

\[
F \supset O \supset A \supset E. \tag{2.15}
\]

### 2.3 Necessary and sufficient conditions of Nash equilibrium for evolutionary games

Sufficient condition of Nash equilibrium in general games have been widely studied in literature [51]. Motivated by this, we explore the necessary and sufficient condition of Nash equilibrium for the evolutionary games in this section. The results not only give a theoretical foundation for the developed algorithms of direct game but also provide a criterion to check whether or not the obtained solution is a Nash equilibrium in practice.

Let \( x = (x_1, \cdots, x_n)^T \in S_n \) be a frequency/adapted strategy vector as discussed above. Denote \( e_i = (0, \cdots, 0, 1, 0, \cdots, 0)^T \) the \( i \)-th unit vector. We have the following theorem on the
necessary and sufficient conditions for \( x \) to be a Nash equilibrium for general payoff function \( \pi(\cdot, \cdot) \) on finite space.

**Theorem 2.3.1. (Necessary and sufficient conditions on finite space)**

A strategy \( x^* \in \mathbb{R}^n \) is a Nash equilibrium, if and only if

1. \( \pi(e_i, x^*) = \pi(x^*, x^*) \) on the support of \( x^* \): \( \text{Spt}(x^*) = \{ i \mid x^*_i > 0 \} \),

2. \( \pi(e_i, x^*) \leq \pi(x^*, x^*) \) on \( \text{Spt}(x^*)^c = \{ i \mid x^*_i = 0 \} \).

**Proof.** If \( x^* \) is a Nash equilibrium, by definition,

\[
\pi(x, x^*) \leq \pi(x^*, x^*),
\]

for all \( x \in \mathbb{R}^n \) with \( \sum_{i=1}^n x_i = 1, \ x_i \geq 0 \) for \( i = 1, 2, \ldots, n \). It is easy to see that \( \pi(e_i, x^*) \leq \pi(x^*, x^*) \) for all \( i \) by definition of Nash equilibrium. Assume there exists \( j \in \text{Spt}(x^*) \) such that \( \pi(e_j, x^*) < \pi(x^*, x^*) \) and it follows that

\[
\pi(x^*, x^*) = \sum_{i: i \neq j} x^*_i \pi(e_i, x^*) + x^*_j \pi(e_j, x^*) < \sum_{i = 1}^n x^*_i \pi(x^*, x^*) = \pi(x^*, x^*)
\]

which leads to contradictions. Hence

\[
\pi(e_i, x^*) = \pi(x^*, x^*), \ \forall \ i \in \text{Spt}(x^*).
\]

On the other hand, for any \( x \in S_n \) we have

\[
\pi(y, y^*) = \sum_{i=1}^n y_i \pi(e_i, y^*)
\]

\[
= \sum_{i: y_i^* > 0} y_i \pi(e_i, y^*) + \sum_{i: y_i^* = 0} y_i \pi(e_i, y^*)
\]

\[
\leq \sum_{i: y_i^* > 0} y_i \pi(y^*, y^*) + \sum_{i: y_i^* = 0} y_i \pi(y^*, y^*)
\]

\[
= \pi(y^*, y^*),
\]

which gives what asserts. \( \square \)
Theorem 2.3.1 can be easily extended to the general case such that the frequency (strategy) denotes some induced measure $\mu_X$ of random variable $X$ and the average payoff for evolutionary game is written into

$$\pi(\mu_X, \nu_X) = \int \int \pi(x, x) d\mu_X d\nu_X. \quad (2.16)$$

The necessary and sufficient condition for $\mu_X$ to define a distribution being a Nash equilibrium is given below. $X$ and $Y$ are both random variables defined on some well defined set $\Omega$.

**Theorem 2.3.2.** *(Necessary and sufficient conditions on general probability measure space)*

Assume $\pi(\mu_X, \nu_X)$ is well defined, then $\mu^*_X$ defines a distribution being a Nash equilibrium, if and only if,

1. $\pi(\mu^*_X, \nu_X) \leq \pi(\mu^*_X, \mu^*_X)$ for all induced probability measures $\nu_X$ restricted on $A \subset \Omega$ such that $\mu^*_X(A) = 0$

2. $\pi(\mu^*_X, \nu_X) = \pi(\mu^*_X, \mu^*_X)$ for all induced probability measures $\nu_X$ restricted on the support of $\mu^*_X$

**Proof.** If $\mu^*_X$ defines a distribution being a Nash equilibrium, $\pi(\mu^*_X, \nu_X) \leq \pi(\mu^*_X, \mu^*_X)$ is obvious by the definition of Nash equilibrium. Assume there exists subsets of support of $\mu^*_X$ on which $\pi(\mu^*_X, \nu_X) < \pi(\mu^*_X, \mu^*_X)$, i.e. $\nu_X(\{\pi(x, \mu^*_X) < \pi(\mu^*_X, \mu^*_X)\}) > 0$ so that

$$\pi(\mu^*_X, \nu_X) = \int \pi(x, \mu^*_X) d\mu^*_X = \int_{\{\mu^*_X > 0\}} \pi(x, \mu^*_X) d\mu^*_X + \int_{\{\mu^*_X = 0\}} \pi(x, \mu^*_X) d\mu^*_X$$

$$= \int_{\{\mu^*_X > 0\} \cap \{\pi(x, \mu^*_X) < \pi(\mu^*_X, \mu^*_X)\}} \pi(x, \mu^*_X) d\mu^*_X$$

$$+ \int_{\{\mu^*_X > 0\} \cap \{\pi(x, \mu^*_X) = \pi(\mu^*_X, \mu^*_X)\}} \pi(x, \mu^*_X) d\mu^*_X$$

$$< \pi(\mu^*_X, \mu^*_X) \cdot \mu^*_X(\Omega)$$

which yields contradiction so that $\nu_X(\{\pi(x, \mu^*_X) < \pi(\mu^*_X, \mu^*_X)\}) = 0$, i.e. $\pi(\mu^*_X, \nu_X) = \pi(\mu^*_X, \mu^*_X)$. 
On the other hand, assume both conditions are satisfied then one has,

\[
\pi(\mu^*_X, \nu_X) = \int \pi(\mu^*_X, x) d\nu_X \\
= \int_{\{\mu^*_X > 0\}} \pi(\mu^*_X, x) d\nu_X + \int_{\{\mu^*_X = 0\}} \pi(\mu^*_X, x) d\nu_X \leq \pi(\mu^*_X, \mu^*_X) \cdot \nu_X(\Omega)
\]

which implies that \( \mu^*_X \) is a Nash equilibrium.

For the evolutionary games defined on finite space with constant payoff matrix \( A \), the following corollary from Theorem 2.3.1 and 2.3.2 is used for developing algorithms for direct games.

**Corollary 2.3.3.** (Necessary and sufficient conditions for algorithm development)
Assume \( \pi(\mu_X, \nu_X) = x^T A x \) where \( x \in S_n \) and \( A \) is a constant payoff matrix, then \( x^* \) is a Nash equilibrium, if and only if,

1. \( e^T i A x^* = (x^*)^T A x^* \) on the support of \( x^* \) : \( Spt(x^*) = \{i \mid x^*_i > 0\} \),
2. \( e^T i A x^* \leq (x^*)^T A x^* \) on \( Spt(x^*)^c = \{i \mid x^*_i = 0\} \).

Moreover, the symmetric evolutionary game has a close connection with the generalized knapsack problem. It can be formulated as to find an optimal solution \( x^* \in S_n \) for the following problem:

\[
\max_{x \in \mathbb{R}^n} \frac{x^T A x}{2} \quad \text{subject to: } \sum_i x_i = 1, \quad x \geq 0.
\] (2.17)

This problem can be considered as a knapsack problem of \( n \) ‘objects’ with the objective function generalized to a symmetric quadratic form \( \frac{x^T A x}{2} \) and with the ‘sack’ restricted in a simplex \( S_n = \{x \in \mathbb{R}^n : x \geq 0, \sum_i x_i = 1\} \) [35]. Consider the ‘objects’ to be the species fractions in a given population and the matrix \( A \) to be the fitness matrix of the species, the objective function for the generalized knapsack problem is exactly half of the average fitness function of the symmetric evolutionary game. Therefore, the goal of the generalized knapsack problem is to maximize the average fitness of the population of the symmetric evolutionary game.
game. By Theorem 2.2.1, an optimal strategy must satisfy a set of complementarity conditions. Correspondingly, an optimal solution to the generalized knapsack problem must also satisfy certain conditions. Start with considering a general constrained optimization problem

$$\min_{x \in \mathbb{R}^n} : f(x)$$

subject to:

$$c_i(x) = 0, \quad i \in E$$

$$c_i(x) \geq 0, \quad i \in I$$

where $f$ is the objective function, $c_i(x)$ the constraints, $E$ the set of indices for equality constraints, and $I$ the set of indices for inequality constraints. Assume that $f$ and $c_i$ are all continuously differentiable. Let $x$ be a feasible solution to the optimization problem. Let $E'(x)$ be the set of indices for the constraints active at $x$, where

$$E'(x) = E \cup \{ i \in I : c_i(x) = 0 \}.$$  \hspace{1cm} (2.19)

Then we have the first order necessary conditions for the optimal solution as stated in the following theorem.

**Theorem 2.3.4.** [29] Let $x^* \in \mathbb{R}^n$ be an optimal solution to the general constrained optimization problem in (2.18). Assume that the gradients of the constraints active at $x^*$, $\{\nabla c_i(x^*) : i \in E'(x^*)\}$, are linearly independent. Then, there must be a set of Lagrangian multipliers $\lambda^* \in \mathbb{R}^{|E|}$, and $\mu^* \in \mathbb{R}^{|I|}$ such that

$$\nabla x L(x^*, \lambda^*, \mu^*) = 0$$

$$c_i(x) = 0, \quad i \in E$$

$$c_i(x) \geq 0, \quad i \in I$$

$$\mu^* \geq 0, \quad i \in I$$

$$\mu^*^T c_i(x^*) = 0, \quad i \in I$$

where $L(x, \lambda, \mu)$ is called the Lagrangian function of (2.18),

$$L(x, \lambda, \mu) = f(x) - \sum_{i \in E} \lambda_i c_i(x) - \sum_{i \in I} \mu_i c_i(x).$$  \hspace{1cm} (2.21)
These conditions are called the KKT conditions of the generalized constrained optimization problem (2.18) named after W. Karush, H. Kuhn, and A. Tucker. Notice that an optimal solution $x^*$ must satisfy the KKT conditions, but a solution $x_k$ that satisfies the KKT condition, called a KKT point, may not always be an optimal solution.

In order to apply Theorem 2.3.4 on the generalized knapsack problem in (2.17), we transform it into a standard minimization problem. First we change the objective function for this problem into

$$ f(x) = -\frac{1}{2}x^TAx. $$

The nonnegative constraints $c_i(x) = x_i \geq 0$, $i = 1, \ldots, n$ and the equality constraint $c_{n+1}(x) = 1 - \sum_i x_i = 0$ provide $n + 1$ constraints. So we have $I = \{1, \ldots, n\}$ and $E = \{n + 1\}$. Let $x$ be a feasible solution for the problem. Let $E'(x)$ be the set of indices for the constraints active at $x$, i.e.,

$$ E'(x) = \{i \in I : c_i(x) = 0\} \cup E \quad (2.22) $$

and $C'(x)$ be the set of gradients of the constraints active at $x$, i.e.

$$ C'(x) = \{\nabla c_i(x) : i \in E'(x)\}. \quad (2.23) $$

Then we have

$$ E'(x) = \{i \in I : x_i = 0\} \cup \{n + 1\} \quad (2.24) $$

and

$$ C'(x) = \{e_i : i \in I, x_i = 0\} \cup \{1\} \quad (2.25) $$

where $e_i$ is the $i$-th unit vector and $1 = \{1, \ldots, 1\}^T$. For any $x \in S_n$, there is at least one $i \in I$ such that $x_i \neq 0$ since $x \geq 0$ and $\sum_i x_i = 1$. Therefore, $C'(x)$ contains the vector $1$ and a subset of vectors $\{e_i : i \in I\}$ which are always linearly independent. We then have the following first-order necessary conditions for the generalized knapsack problem:

**Theorem 2.3.5.** Let $A \in \mathbb{R}^{n \times n}$ be a symmetric fitness matrix and $S_n = \{x \in \mathbb{R}^n : x \geq 0, \sum_i x_i = 1\}$ the set of all feasible solutions for the generalized knapsack problem in (2.17). If $x^* \in S_n$ is an optimal solution, then there must be a scalar $\lambda^*$ such that

$$ x_i^* \geq 0, \quad \lambda^* - (Ax^*)_i \geq 0, \quad (2.26) $$

$$ x_i^*(\lambda^* - (Ax^*)_i) = 0, \quad i = 1, \ldots, n. \quad (2.27) $$
Proof. The Lagrangian function for the generalized knapsack problem in (2.17) can be written as:

\[ L(x, \lambda, \mu) = -\frac{x^T Ax}{2} - \lambda(1 - \sum_i x_i) - \mu^T x, \]

where \( x \in \mathbb{R}^n \), \( \lambda \in \mathbb{R} \), and \( \mu \in \mathbb{R}^n \). Since for this problem the gradients of the active constraints at any \( x \in S_n \) are linearly independent, by Theorem 2.3.4, if \( x^* \in S_n \) is an optimal solution to the generalized knapsack problem in (2.17), then there must be \( \lambda \in \mathbb{R} \), and \( \mu \in \mathbb{R}^n \) such that

\[
\nabla_x L(x^*, \lambda^*, \mu^*) = -Ax^* + \lambda^* 1 - \mu^* = 0,
\]

\[
\sum_i x_i^* = 1,
\]

\[
x^* \geq 0,
\]

\[
\mu^* \geq 0,
\]

\[
x^{*T} \mu^* \geq 0.
\]

From the Lagrangian function we have \( \mu^* = -Ax^* + \lambda^* 1 \). It follows for \( x^* \in S_n \),

\[
x^* \geq 0,
\]

\[
\lambda^* 1 - Ax^* \geq 0,
\]

\[
x^{*T}(\lambda^* 1 - Ax^*) \geq 0.
\]

\[\square\]

Note that the conditions in Theorem 2.2.1 and Theorem 2.3.5 are the same. However, it does not imply that the symmetric evolutionary game in (9.1) is equivalent to the generalized knapsack problem in (2.17), because the conditions are necessary and sufficient for an optimal strategy for (9.1) but only necessary for an optimal solution for (2.17). Therefore, an optimal solution for the generalized knapsack problem in (2.17) must be an optimal strategy for the symmetric evolutionary game in (9.1), while the converse may not necessarily be true. We state this conclusion as a corollary from Theorem 2.2.1 and Theorem 2.3.5 in the following.

**Corollary 2.3.6.** An optimal solution \( x^* \in S_n \) for the generalized knapsack problem in (2.17) must be an optimal strategy for the symmetric evolutionary game in (9.1), while an optimal
strategy $x^* \in S_n$ for the symmetric evolutionary game in (9.1) is only a KKT point for the generalized knapsack problem in (2.17), which is necessary but not sufficient for $x^*$ to be optimal for the generalized knapsack problem in (2.17).

Furthermore, we focus on the generalized knapsack problem in (2.17) and derive additional second order necessary and sufficient conditions for its optimal solutions.

Again, first consider the general constrained optimization problem in (2.18). Let $x^*$ be an optimal solution. Let $E'(x^*)$ be the set of indices for the constraints active at $x^*$,

$$E'(x^*) = E \cup \{i \in I : c_i(x^*) = 0\}$$

and $C'(x^*)$ be the Jacobian of the constraints active at $x^*$,

$$C'(x^*) = \{\nabla c_i(x^*) : i \in E'(x^*)\}^T.$$  \hfill (2.29)

The second order necessary and sufficient conditions for an optimal solution to the general constrained optimization problem in (2.18) are summarized as following.

**Theorem 2.3.7.** Let $x^* \in \mathbb{R}^n$ be an optimal solution to the general constrained optimization problem in (2.18). If $C'(x^*)$ has full row rank $m$, let $Z \in \mathbb{R}^{n \times (n-m)}$ be the null space matrix of $C'(x^*)$, then

$$y^T Z^T \nabla^2 f(x^*) Z y \geq 0, \quad \text{for all} \quad y \in \mathbb{R}^{n-m}, \; y \neq 0,$$

i.e., the reduced Hessian of $f(x)$ at $x^*$ must be positive semi-definite.

Now we consider a KKT point $x^* \in \mathbb{R}^n$ for the general constrained optimization problem in (2.18). Let $E'(x^*)$ be the set of indices for the constraints strongly active at $x^*$, i.e.

$$E'(x^*) = E \cup \{i \in I : c_i(x^*) = 0\},\hfill (2.31)$$

where $\lambda_i^*$ is the Lagrangian multiplier in the KKT conditions for $x^*$. We have the second order necessary and sufficient conditions for $x^*$ to be a strictly optimal solution to the general constrained optimization problem in (2.18), i.e. $f(x^*) < f(x)$ for all $x$ in some neighborhood of $x^*$ and $x \neq x^*$.  

Theorem 2.3.8. Let \( x^* \in \mathbb{R}^n \) be a KKT point to the general constrained optimization problem in (2.18). Assume that \( C'(x^*) \) has full row rank \( m \). Let \( Z \in \mathbb{R}^{n \times (n-m)} \) be the null space matrix of \( C'(x^*) \). If
\[
y^T Z^T \nabla^2_x f(x^*) Z y > 0, \quad \text{for all } y \in \mathbb{R}^{n-m}, \ y \neq 0,
\]
i.e., the reduced Hessian of \( f(x) \) at \( x^* \) is positive semi-definite, then \( x^* \) must be a strictly optimal solution to the general constrained optimization problem in (2.18).

Now apply Theorem 2.3.7 and Theorem 2.3.8 to the generalized knapsack problem in (2.17). With the similar transformation in Theorem 2.3.5, let \( E'(x^*) \) be the set of indices for the constraints active at \( x^* \). \( C'(x^*) \) has the vector \( 1^T \) and a subset of vectors \( \{ e_i^T : i \in I \} \) as the rows, and is of full row rank. The hessian of the objective function \( f(x) = -x^T A x / 2 \) is \( \nabla^2_x f(x^*) = -A \). Then we have the second order necessary conditions for \( x^* \) to be an optimal solution to the generalized knapsack problem in (2.17).

Theorem 2.3.9. Let \( x^* \in S_n \) be an optimal solution to the generalized knapsack problem in (2.17). Let \( C'(x^*) \) has full row rank \( m \). Let \( Z \in \mathbb{R}^{n \times (n-m)} \) be the null space matrix of \( C'(x^*) \). Then
\[
y^T Z^T A Z y \leq 0, \quad \text{for all } y \in \mathbb{R}^{n-m}, \ y \neq 0,
\]
i.e., the reduced Hessian of the objective function \( f(x) = -x^T A x / 2 \) at \( x^* \) must be negative semi-definite.

Now consider a KKT point \( x^* \in S_n \). Let \( E'(x^*) \) be the set of indices for the constraints strongly active at \( x^* \). Then \( C'(x^*) \) has the vector \( 1^T \) and a subset of vectors \( \{ e_i^T : i \in I \} \) as the rows, and is of full row rank. Also the hessian of the objective function is \( \nabla^2_x f(x^*) = -A \). Then we have the second order sufficient conditions for \( x^* \) to be a strictly optimal solution to the generalized knapsack problem in (2.17).

Theorem 2.3.10. Let \( x^* \in S_n \) be a KKT point to the generalized knapsack problem in (2.17). Let \( C'(x^*) \) has full row rank \( m \). Let \( Z \in \mathbb{R}^{n \times (n-m)} \) be the null space matrix of \( C'(x^*) \). Then
\[
y^T Z^T A Z y < 0, \quad \text{for all } y \in \mathbb{R}^{n-m}, \ y \neq 0,
\]
i.e., if the reduced Hessian of the objective function $f(x) = -\frac{x^TAx}{2}$ at $x^*$ is negative definite, then $x^*$ must be a strictly optimal solution to the generalized knapsack problem in (2.17).

Theorem 2.3.9 and 2.3.10 can be very useful for checking the optimality of the solutions for the generalized knapsack problems and hence the strategies for the symmetric evolutionary games beyond the conditions given in Theorem 2.2.1 and 2.3.4. All we need to do is to find the null space matrix $Z$ and the eigenvalues of the reduced Hessian $Z^TAZ$ to see if it is negative semi-definite or negative definite. For example, suppose that we have a KKT point $x^* \in S_n$ for the generalized knapsack problem in (2.17) at which the only active constraint is the equality constraint $\sum_i x_i = 1$. Then, $C'(x^*) = -1^T$, for which we can construct a null space matrix $Z \in \mathbb{R}^{n \times (n-1)}$ such that $Z_{ij} = 0$ for $i \neq j, j + 1$, and $Z_{ii} = 1$, $Z_{i+1,j} = -1$.

In general, we can use for example the QR factorization to find the null space matrix [45]. Then the optimality of $x^*$ can be tested by checking the eigenvalues of the reduced Hessian $Z^TAZ$. If any of the eigenvalues is positive, $x^*$ is not optimal, and if all the eigenvalues are negative, $x^*$ must be optimal and even strictly optimal. This is applied to a solver that will be discussed in chapter 8.
CHAPTER 3. INVERSE GAME BASED ON REPLICATOR DYNAMICS

3.1 Introduction

If we have the dynamics of the replicator equations system, then we might ask the question: ‘given the data that we have available, what can we say about the corresponding games?’ The solution to this problem is useful because it generally tells us something about a physical parameter that we cannot directly observe. Thus, inverse games are important. However, the inverse games are not well studied in evolutionary game theory.

An inverse game targets on recovering payoff matrix $A$ for the evolutionary game model based on the data and replicator equations (2.9), whose parameters are the components of the payoff matrix. To obtain the estimation and inference on $A$, we have four steps: data transformation, data smoothing, estimation based on nonlinear regression, and sub-sampling based on measurement error model. Start with unifying the different types of data sets, we then use non-parametric spline methods to estimate the derivatives. With the smoothed data, we apply the least squares method to obtain the estimation of the payoff matrices, and we use the parametric bootstrap sampling method to obtain the inferences of the payoff matrices.

3.2 Data transformation

The observed data set for inverse game is usually in the form of time series from either experiments or observational studies. Some of the data set is consistent with the domain of replicator equations that the data values are in the form of frequency and sum up to 1 at each observation time point such as [47], while others may be in the form of predator-prey type that the data value is nonnegative real values which are consistent with Lotka-Volterra
equations. To unify the different types of data set for the purpose of estimating the payoff matrix, we establish the topological equivalence of replicator systems and standard Lotka-Volterra system as suggested in [30]. We start with the following proposition.

**Proposition 3.2.1.** Consider the replicator dynamics (2.9) associated with the payoff $A$. The addition of a constant $d_j$ to the $j$th column of payoff matrix $A$ does not change (2.9) on $S_n$. Hence, by adding appropriate constants, $A$ may be transformed into simpler form with all elements on the last row equal to 0.

**Proof.** W.L.O.G., adding $d_j \in \mathbb{R}$ to the $j$th column of $A$, so that $\tilde{A} = (a_{11}, \cdots, a_{ij} + d_j \cdot 1, \cdots, a_{nn})$. The replicator dynamics (2.9) is therefore rewritten as, for $i \in \{1, \cdots, n\}$

\[
\frac{dx_i}{dt} = x_i \left( (\tilde{A}x)_i - x^T \tilde{A}x \right)
\]

\[
\frac{dx_i}{dt} = x_i \left\{ \sum_{k \neq j, k=1}^{n} a_{ik} x_k + (a_{ij} + d_j)x_j \right\} - x_i \left\{ \sum_{l=1}^{n} \sum_{k=1, k \neq j}^{n} a_{lk} x_l x_k + \sum_{l=1}^{n} (a_{lj} + d_j)x_l x_j \right\}
\]

\[
= x_i \left( (Ax)_i - x^T Ax \right) + d_j x_i x_j - d_j x_i x_j \left( \sum_{i=1}^{n} x_i \right)
\]

\[
= x_i \left( (Ax)_i - x^T Ax \right) + d_j x_i x_j - d_j x_i x_j \left( \sum_{i=1}^{n} x_i \right)
\]

as $x \in S_n$. Hence, the replicator dynamics is invariant under translation by adding constants on each column. Furthermore the replicator dynamics defined by $A$ is equivalent to the system whose payoff matrix has all elements equal to zeros in the last row. \[
\]

Using Proposition 3.2.1, we have the topological equivalence established as following.

**Theorem 3.2.2.** There exists a diffeomorphism from $S_n^+ = \{x \in S_n \mid x_n > 0\}$ onto $\mathbb{R}^{n-1}_+$ mapping the orbits of the replicator dynamics defined by (2.9) onto the orbits of the Lotka-Volterra dynamics defined by

\[
\frac{dy}{dt} = y^T (r + B y)
\]

where for $A = [a_{ij}]$, $B := [b_{ij}] = a_{ij} - a_{nj}$ is an $(n-1) \times (n-1)$ matrix and $r_i = a_{in} - a_{nn}$ for $i \in \{1, \cdots, n-1\}$. 

Proof. Let \( y = \frac{1}{n-1} \sum_{i=1}^{n-1} y_i \). Define the transformation by

\[
x_i = \frac{y_i}{1+y}
\]

(3.2)

for \( i \in \{1, \cdots, n-1\} \) and

\[
x_n = \frac{1}{1+y}
\]

(3.3)

that maps hyperplane \( \{ y \in \mathbb{R}_+^n \mid y_n = y \equiv 1 \} \) onto \( S_n \). It immediately follows the definition that the inverse transform is given by

\[
y_i = \frac{x_i}{x_n}
\]

(3.4)

for \( i \in \{1, \cdots, n\} \).

Consider the replicator dynamics given by (2.9). By Proposition 3.2.1, w.l.o.g, we may assume that the last row of \( A \) is zero row vector. The transformation (3.4) implies for \( i \in \{1, \cdots, n-1\} \)

\[
\frac{dy_i}{dt} = \frac{d}{dt} \left( \frac{x_i}{x_n} \right) = \left( \frac{x_i}{x_n} \right) \{(Ax)_i - (Ax)_n\} = \left( \frac{x_i}{x_n} \right) (Ax)_i
\]

(3.5)

by the assumption on \( (Ax)_n \equiv 0 \). Hence, we have

\[
\frac{dy_i}{dt} = y_i \left\{ \sum_{j=1}^{n} a_{ij}x_j \right\} = y_i \left\{ \sum_{j=1}^{n} a_{ij}y_j \right\} x_n
\]

(3.6)

for \( i \in \{1, \cdots, n-1\} \).

Finally, let’s consider a new time scale defined by \( \tau := \phi(t) \) for some \( \phi \) to be determined. (3.6) implies

\[
\frac{dy_i}{dt} = y_i \left\{ a_{in} + \sum_{j=1}^{n-1} a_{ij}y_j \right\}
\]

(3.8)
since \( y_n \equiv 1 \), and therefore the Lotka-Volterra equation (3.1) as desired. Similarly, we can
derive the transformation from (3.1) to (2.9). The differentiability and invertibility of the
transformation between (2.9) and (3.4) is obvious due to its definition. We therefore obtain
the assertion.

Hence, for the purpose of data manipulation in practice, we have the following corollary.

**Corollary 3.2.3.** There exists a differentiable and invertible transformation to convert data
of Lotka-Volterra type to data of replicator type and vice versa.

Assume that \( \{y(t_1), \ldots, y(t_m)\} \) are the data of predator-prey type that is consistent with
the Lotka-Volterra dynamics (3.1). Using the proof of Theorem 3.2.2, we have time re-scaling
defined by
\[
\tau := \psi(t) = \int_0^t \frac{1}{x_n(t)} \, dt.
\]
W.l.o.g, assume \( t_0 = \tau_0 = 0 \), the new time points are defined by
\[
\tau_j = \int_0^{t_j} 1 + y(s) \, ds = t_j + \int_0^{t_j} y(s) \, ds
\]
for \( j = 1, 2, \ldots, m \), where the integral term can be approximated using spline or local poly-
nomial regression in practice.

Finally, using the transformation defined by (3.2) and (3.3), we have
\[
x_i(\tau_j) = \frac{y_i(\tau_j)}{1 + y(\tau_j)}, \quad \text{and} \quad x_n(\tau_j) = \frac{1}{1 + y(\tau_j)}
\]
where \( y(\tau_j) := \sum_{i=1}^{n-1} y_i(\tau_j) \) for \( j = 1, 2, \ldots, m \).

It is analogous to covert data consistent with replicator dynamics \( \{x(t_1), \ldots, x(t_m)\} \) into
the data of predator-prey type. This transformation is useful for the statistical inference of
the payoff matrices via parametric bootstrap below.

### 3.3 Smoothing and least squares

With the data in the appropriate form in hand, we use nonparametric spline methods to
estimate the derivatives \( \frac{\dot{x}(t)}{dt} \) (or \( x'(\tau) \) is necessary). The term nonparametric means we do
not make any assumptions on the measurement distribution at each time point at this step, but rather use a spline interpolating the data to obtain a smooth curve for the purpose of estimating derivatives. Adapting cubic splines [5; 12] we obtain the estimates of derivatives by

\[
\hat{x}'(t_k) = S'(x)(t_k)
\] (3.12)

where \(S'(\cdot)\) denotes the derivative of cubic spline interpolating data \(x\) with respect to coordinate \(t\). The estimated data values are \(S(x)(t_i)\) where \(k = 1, 2, \cdots, K\) with \(K \geq n^2\).

With the data and its derivatives smoothed and estimated by (3.12), the point estimates of payoff matrix \(A\) is obtained by penalized least squares method that

\[
\hat{A}_\eta := \arg\min_{A \in \Omega_A} \sum_{k=1}^K \left( \hat{x}'(t_k) - F(\hat{x}(t_k), A) \right)^2 + \eta ||\text{vec}(A)||_F^q
\] (3.13)

where \(F(\hat{x}(t_k), A)\) is the right hand sides of (2.9), \(\Omega_A \subset \mathbb{R}^{n \times n}\), smoothing parameter \(\eta \in \mathbb{R}^+\) is always assumed to be small, and vec is the vectorization operator. The penalized least squares defined by (3.13) is in fact of the form of lasso if \(q = 1\) and of the form of ridge regression if \(q = 2\) [5]. It is known that [5], \(\hat{A}_\eta\) is a biased but consistent estimator of \(A\) and provides a small MSRE in practice.

### 3.4 Inferences based on parametric bootstraps

Least squares method based on spline smoothing of the appropriate data set provides the point estimation of \(A\). To obtain the inferences of \(A\), i.e. the 95% credible intervals, standard deviation of the estimates, biases or the MSRE of the estimates, we use the well-known parametric bootstrap sampling method [7] with assumption that the data satisfied measurement error statistical model.

The procedure is discussed as following. Let \(A_\eta^* := \hat{A}_\eta\) with pre-specified \(\eta\). Then we generate data of replicator type based on \(A_\eta^*\) and the replicator equation (2.9) to get \(\{x_\eta^*(t_k)\}_{k=1}^K\) and transform them into predator-prey type to get \(\{y_\eta^*(\tau_j)\}_{j=1}^K\). Using the measurement error statistical model, we assume

\[
y_\eta(t_j) = y_\eta^*(t_j) + \epsilon_j,
\] (3.14)
where $\epsilon_j \sim i.i.d. \text{MVN}_n(0, \Sigma)$ where $\Sigma$ is some pre-specified covariance matrix representing the measurement error at each measurement time. In practice, $\Sigma = \sigma^2 I_{n \times n}$ where $\sigma^2$ denotes the measurement variance.

By sampling $\epsilon_j$ from $\text{MVN}_n(0, \Sigma)$ we can obtain $B$ subsamples of $\left\{\tilde{Y}(\tau_j)\right\}_{j=1}^K$, $b = 1, 2, \cdots, B$, and simultaneously $B$ subsamples of $\left\{\tilde{X}(t_k)\right\}_{k=1}^K$, $b = 1, 2, \cdots, B$ by the method discussed in section (3.2). Repeatedly performing the smoothing and least squares method on the subsamples, we obtain the parametric bootstrap samples of $\tilde{A}_1^n, \tilde{A}_2^n, \cdots, \tilde{A}_B^n$, and then derive the inferences on $A$ such as standard deviations, biases, 95% credible intervals, MSRE etc. using sample statistics [7].

Using $\tilde{A}_1^n, \tilde{A}_2^n, \cdots, \tilde{A}_B^n$, we can compute the 95% credible bands of the original data set as well, by which we can adjust $\eta$ to reduce the bias of the estimation of $A$.

The scheme for the inverse evolutionary game is therefore summarized as below:

**Algorithm 3.4.1.** (Inverse game scheme)

- **Step 1. Transformation:** Convert whole data set, $\left\{\left\{y_i(t_j)\right\}_{i=1,j=1}^{n,m}, \{t_j\}_{j=1}^m\right\}$ into the form consistent with the replicator dynamics if necessary using the results from section (3.2).

- **Step 2. Smoothing:** Using spline to smooth the data and estimate both the derivatives and state variable

  $\left\{\hat{x}'(t_k)\right\}_{k=1}^K$, and $\left\{\hat{x}(t_k)\right\}_{k=1}^K$,

  where $x = (x_1, \cdots, x_n)^T$.

- **Step 3. Estimation based on nonlinear regression:** Estimation of the payoff matrix $A_{n \times n}$ on $\mathbb{R}^{n \times n}$ based on the replicator dynamics using penalized least squares methods in (3.13).

- **Step 4. Subsampling based on measurement error model:** Using parametric bootstrap procedure discussed in section (3.4) to derive the inferences on $A$ such as biases, standard deviations, 95% credible intervals, MSRE etc. using sample statistics.
CHAPTER 4. DIRECT EVOLUTIONARY GAMES

4.1 Introduction

By Nash’s theorem, we know that every finite game has at least one mixed Nash equilibrium [28]. Equilibrium computation has been one of the earliest research goals of algorithmic game theory. One of the most basic questions is: given a bimatrix game with integer payoffs, find one Nash equilibrium of the game. It was recently shown that this problem is ‘PPAD-complete’ [37].

For evolutionary games, we first consider the classical Snow-Shapley method and Lemke-Howson method. The classical Snow-Shapley procedure [40] is an algorithm that exhausts all subsystems. The Lemke-Howson algorithm [18] is a classical simplex type algorithm developed to search Nash equilibria of two-player, finite-strategy games. By making necessary assumptions, we specialize the classical methods for evolutionary games. Also by directly using the theoretical results from chapter 2, we propose an algorithm based on the solution of a complementarity problem on a simplex to solve the direct evolutionary games, i.e. search for the Nash equilibria.

4.2 Specialized Snow-Shapley algorithm

The Snow-Shapley procedure [40] is a classical algorithm for finding all extreme optimal strategies via exhausting all subsystems and is a purely combinatorial algorithm. For the evolutionary game, we prove a specialized Snow-Shapley type theorem in following.

**Theorem 4.2.1.** (Snow-Shapley type theorem for evolutionary games) Given an evolutionary game defined by matrix $A \in \mathbb{R}^{n \times n}$, $x^* \in S_n$ is a Nash equilibrium if and only if there is a
submatrix $M \in \mathbb{R}^{r \times r}$ of $A$, which is obtained by removing rows and columns of $A$ with the same indices, and $1_r^T \text{adj}(M) 1_r \neq 0$, where $1_r = (1, \ldots, 1)^T$ such that

$$x_r^* = \frac{\text{adj}(M) 1_r}{1_r^T \text{adj}(M) 1_r}$$

(4.1)

where $x_r^*$ is the $r$-dimensional vectors obtained by deleting from $x^*$ the components corresponding to the rows and columns of $A$ that must be removed to construct $M$, and

$$x^T A x^* \geq e_i^T A x^* \quad \text{for all} \quad i \in \text{Spt}(x^*)^c. \quad \text{(4.2)}$$

Proof. First of all, we compute the average payoff $x^T A x^*$ if (4.1) is satisfied. Using elementary row and column operation, we can decompose $A$ such that

$$A = \begin{pmatrix} M & C \\ D & E \end{pmatrix}$$

(4.3)

where $M$ corresponds to submatrix with rows and columns having same original indices in $A$ according to the nonzero components of $x^*$, and $C, D, E$ are submatrices of $A$. We can also arrange $x^* = (x_r^T, 0_{n-r}^T)^T$ so that

$$x^T A x^* = (x_r^T, 0_{n-r}^T) \begin{pmatrix} M & C \\ D & E \end{pmatrix} \begin{pmatrix} x_r^* \\ 0_{n-r} \end{pmatrix}$$

$$= x_r^T M x_r^*$$

$$= \frac{1_r^T (\text{adj}(M))^T M \cdot \text{adj}(M) 1_r}{(1_r^T \text{adj}(M) 1_r)^2}$$

$$= \frac{\det(M) 1_r^T (\text{adj}(M))^T 1_r}{(1_r^T \text{adj}(M) 1_r)^2}$$

$$= \frac{\det(M)}{1_r^T \text{adj}(M) 1_r}$$

(4.4)

by the fact that $M \text{adj}(M) = \text{adj}(M) M = \det(M) I_{r \times r}$. 


Next, for all $i \in Spt(x^*)$, using elementary row and column operation again we have

$$e_i^T Ax^* = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ \vdots \\ 0_{n-r} \end{pmatrix}^T \begin{pmatrix} M & C \\ D & E \end{pmatrix} \begin{pmatrix} x_r^* \\ 0_{n-r} \end{pmatrix} = (0 \cdot 0, 0, 0, \ldots, 1 \cdots, 0) \cdot 1_r,$$

$$= \frac{\det(M)_{1r} \text{adj}(M)_{1r}}{1_r \text{adj}(M)_{1r}} \cdot \begin{pmatrix} 0, 0, 0, \ldots, 1 \cdots, 0 \end{pmatrix}_{1_r} \cdot 1_r = \begin{pmatrix} 0, 0, 0, \ldots, 1 \cdots, 0 \end{pmatrix}_{1_r} \cdot 1_r.$$

Hence, $x^T A x^* = e_i^T A x^*$ for $i \in Spt(x^*)$; and for all $i \in Spt(x^*)^c$, the assumption (4.2) guarantees that $x^T A x^* \geq e_i^T A x^*$. By Corollary 2.3.3, we conclude that $x^*$ defined by (4.1) and (4.2) is a Nash equilibrium.

On the other hand, consider $x^*$ is a Nash equilibrium. First of all, let’s assume $Spt(x^*)^c \neq \emptyset$. By Corollary 2.3.3, (4.2) is true for $i \in Spt(x^*)^c$ and $x^T A x^* = e_i^T A x^*$ for all $i \in Spt(x^*)$. For each $i \in Spt(x^*)$, employing the elementary row and column operations, we have

$$x^T A x^* = e_i^T A y^*$$

$$= (x_r^T, 0_{n-r}^T) \begin{pmatrix} M & C \\ D & E \end{pmatrix} \begin{pmatrix} x_r^* \\ 0_{n-r} \end{pmatrix} = (0, 0, 0, \ldots, 1 \cdots, 0)_r \cdot M x_r^* \cdot 1_r.$$

Assume $x_r^T M x_r^* \neq 0$ in (4.6), then $x^* = (x_r^T M x_r^*)^{-1} 1_r$. As $x^* \in S_n$, we have

$$1_r^T x_r^* = 1$$

$$x_r^T M x_r^* = \frac{1}{1_r^T M^{-1} 1_r}$$

$$x^* = \frac{M^{-1} 1_r}{1_r^T M^{-1} 1_r} = \frac{\text{adj}(M) 1_r}{1_r^T M^{-1} 1_r}.$$

(4.7)
as $M^{-1} = \det(M)^{-1}\adj(M)$.

It remains to consider the case that $x_r^*^T M x_r^* = 0$ in (4.6). Suppose $x_r^*^T M x_r^* = 0$. By Proposition 3.2.1, adding a positive constant $c$ to each column of $A$ results in a strategically equivalent game $A_c$ with $x_r^*^T M x_r^* > 0$. Repeating above argument for $A_c$, we have $x^*$ is a Nash equilibrium if and only if there is an $r \times r$ submatrix $M_c = M + cJ_r$ of $A_c$ where $J_r$ is an $r \times r$ matrix with all components equal to 1 and with $1_r^T \adj(M_c) 1_r \neq 0$ such that

$$x_r^* = \frac{\adj(M_c) 1_r}{1_r^T \adj(M_c) 1_r}$$

where $x_r^*$ is an $r$-dimensional vectors obtained by deleting from $x^*$ the components corresponding to the rows and columns of $A_c$ with same original indices, which must be deleted to construct $M_c$. Recall the facts that

$$\adj(M + cJ_r) 1_r = \adj(M) 1_r$$

and

$$\det(M + cJ_r) = \det(M) + c1_r^T \adj(M) 1_r,$$

$1_r^T \adj(M_c) 1_r \neq 0$ if and only if $1_r^T \adj(M) 1_r \neq 0$, and therefore (4.8) is true for $M_c$ if and only if (4.1) is true for $M$ with $x_r^*^T M x_r^* = 0$.

Finally, if $\text{Spt}(x^*)^c = \emptyset$ then $x^*$ is a mixed strategy, replacing all $M$ in the above arguments by $A$ automatically leads to the same conclusion.

We therefore have the proof completed. $\square$

Based on the Theorem 4.2.1, it is easy to see that the complexity of specialized Snow-Shapley algorithm for evolutionary game is

$$\sum_{1 \leq m \leq n} C_n^m m^5 = \sum_{1 \leq m \leq n} \frac{n!m^3}{m!(n-m)!},$$

(4.9)

since the adjoint matrix $\adj(M)$ need to be calculated for each submatrix of $A$ and $M^{-1} = \det(M)^{-1}\adj(M)$. The algorithm is summarized as

Algorithm 4.2.2. (Specialized Snow-Shapley algorithm for evolutionary game)

Input: Payoff matrix $A$.

Output: Complete set of optimal strategies, i.e. Nash equilibria.
• Step 1. Examine the submatrix \( M \), which is obtained by removing the same \( i_1, \ldots, i_r \) rows and columns from payoff matrix \( A \); discard this submatrix if it violates the condition in Theorem 4.2.1;

• Step 2. Compute the solution \( x^*_r \) using the (4.1), and discard the solution and submatrix if \( x^*_r \) has negative component;

• Step 3. Examine the solution \( x^* \) consist of \( x^*_r \) and \( 0_{n-r} \), discard the solution if it violates the necessary and sufficient conditions in Theorem 4.2.1;

• Step 4. All the legitimate solution will form the set of Nash equilibriums of evolutionary game with payoff matrix \( A \).

4.3 Specialized Lemke-Howson algorithm

The Lemke-Howson algorithm [18] is a classical simplex type algorithm developed to search Nash equilibria of two-player, finite-strategy games. It takes two payoff matrices \( A \in \mathbb{R}^{M \times N} \) and \( B \in \mathbb{R}^{M \times N} \) while the player I has the set of choices \( S_1 = \{x_1, x_2, \ldots, x_m\} \) and the player II has the set of choices \( S_2 = \{y_1, y_2, \ldots, y_n\} \). Assume that the player I picks rows and the player II picks columns. With mixed-strategies \((x, y)\), the payoff for player I is \( x^T A y \) and for player II is \( x^T B y \). Let \( M = \{1, 2, \ldots, m\} \) and \( N = \{m + 1, m + 2, \ldots, m + n\} \). Define the support of \( x \) by \( \text{Supp}(x) = \{i| x_i > 0\} \). Define the support of \( y \) similarly.

Definition 4.3.1. A bimatrix game \((A, B)\) is non-degenerate if and only if for every strategy \( x \) of the row player, \( |\text{Supp}(x)| \) is at least the number of pure best responses to \( x \), and for every strategy \( y \) of the column player, \( |\text{Supp}(y)| \) is at least the number of pure best responses to \( y \).

An equivalent definition is: for any \( y' \) that is a best response to \( x \), \( |\text{Supp}(x)| \geq |\text{Supp}(y')| \), and for any \( x' \) that is a best response to \( y \), \( |\text{Supp}(y)| \geq |\text{Supp}(x')| \). Also note that we can slightly perturb the payoff matrices to make the game non-degenerate. Therefore with little loss of generality, we can assume that game \((A, B)\) is non-degenerate. The following proposition is directly implied by the definition:
Proposition 4.3.2. If \((x, y)\) is a Nash equilibrium of a non-degenerate bimatrix game, then 
\(|\text{Supp}(x)| = |\text{Supp}(y)|\).

Now consider the following Polytopes:
\[ P = \{ (u, x) | x_i \geq 0, \sum x_i = 1, x^T B \leq u1 \}, \]
\[ Q = \{ (v, y) | y_i \geq 0, \sum y_i = 1, Ay \leq v1 \}. \]

By the above proposition it is easy to see that every Nash equilibrium can be described as a pair of corner points of \(P\) and \(Q\). For simplicity of notation, consider the following transformations:
\[ P' = \{ x | x_i \geq 0, x^T B \leq 1 \}, \]
and
\[ Q' = \{ y | y_j \geq 0, Ay \leq 1 \}. \]

There is a one to one correspondence between the corners of \(P\) and \(P'\), except the zero corner of \(P'\). In fact, for each corner \((u, x)\) of \(P\), \(\frac{x}{u}\) is a corner of \(P'\); and for each nonzero corner \(x\) of \(P'\), \(\left(\frac{1}{\sum x_i}, \frac{x}{\sum x_i}\right)\) is a corner of \(P\). The same correspondence exists for \(Q\) and \(Q'\).

The corner points of \(P'\) and \(Q'\) are of our interest because they correspond to special set of strategies of the players. \(x\) is a corner point of \(P'\) implies some inequalities among \(\{x | x_i \geq 0, x^T B \leq 1 \}\) bind. If \(x_i = 0\), then row \(i\) is not used in the mixed strategy \(x\); if \((x^T B)_j = 1\), then column \(j\) is a best response to row players strategy \(x\). Next we give an explicit connection of the corner points of \(P', Q'\) and Nash equilibria.

Define graph \(G_1, G_2\) as following: the vertices of \(G_1, G_2\) are the corner points of \(P', Q'\) respectively. There is an edge between \(x_1\) and \(x_2\) in \(G_1\) if and only if \(x_1\) and \(x_2\) are adjacent corner points of \(P'\). Define the edges of \(G_2\) similarly. Then label each vertex \(x\) of \(G_1\) with the indices of the tight constraints in \(P'\), i.e.
\[ L(x) = \{ i | x_i = 0 \} \cup \{ j | (x^T B)_j = 1 \}. \]
Label \(G_2\) similarly. By the non-degeneracy of the game, \(|L(x)| \leq m\) and \(|L(y)| \leq n\). We have the following theorem.
Theorem 4.3.3. A pair \((x, y)\) is a Nash equilibrium if and only if \((x, y)\) is completely labeled:

\[ L(x) \cup L(y) = M \cup N = \{1, 2, \ldots, m + n\} \]

Proof. Suppose \(L(x) \cup L(y) = \{1, 2, \ldots, m + n\}\). For each \(i \in M\) that is in the label set of \(x\), row \(i\) is not used in \(x\), for each \(j \in N\) that is in the label set of \(x\), column \(j\) for the other player is a best response to \(x\). These conclusions are symmetric for the label set of \(y\).

Let \(M_1 = \{i| x_i = 0\}, N_2 = \{j| (x^T B)_j = 1\}; N_1 = \{j| y_j = 0\}, M_2 = \{i| (A y)_i = 1\}\). Since \(|L(x)| \leq m\) and \(|L(y)| \leq n\), then \(L(x) \cup L(y) = M \cup N\) implies \((M_1, M_2)\) is a partition of \(M\) and \((N_1, N_2)\) is a partition of \(N\). Therefore \(x\) consists of strategies only in \(M_2\), and is a best response to \(y\), \(y\) consists of strategies only in \(N_2\) and is a best response to \(x\).

On the other hand, if \((x, y)\) is a pair of Nash equilibrium, then \(M \setminus \text{Supp}(x) \subseteq L\) because those rows are not used in \(x\), and \(\text{Supp}(y) \subseteq L\) because those columns are best responses to \(x\). Note the game is nondegenerate, so \(|\text{Supp}(x)| = |\text{Supp}(y)|\), then \(L(x) = (M \setminus \text{Supp}(x)) \cup \text{Supp}(y)\). Similarly, \(L(y) = (N \setminus \text{Supp}(y)) \cup \text{Supp}(x)\). Hence \(L(x) \cup L(y) = M \cup N\). \(\Box\)

Finally, we use this connection of Nash equilibrium and graphs \(G_1, G_2\) to give the Lemke and Howson algorithm. The basic idea is to pivot alternatively in \(P'\) and \(Q'\) until we find a pair that is completely labeled.

Let \(G = G_1 \times G_2\), i.e., vertices of \(G\) are defined as \(v = (v_1, v_2)\) where \(v_1 \in V(G_1)\) and \(v_2 \in V(G_2)\). There is an edge between \(v = (v_1, v_2)\) and \(v' = (v'_1, v'_2)\) in \(G\) if and only if \((v_1, v'_1) \in E(G_1)\) or \((v_2, v'_2) \in E(G_2)\). Then for each vertex \(v = (v_1, v_2) \in V(G)\), define its label by \(L(v) = L(v_1) \cup L(v_2)\). For each \(k \in M \cup N\), define the set of ‘\(k\)-almost’ completely labeled vertices by

\[ U_k = \{v \in V(G)| L(v) \supseteq M \cup N \setminus \{k\}\}. \]

We have the following key results of \(U_k\):

Theorem 4.3.4. For any \(k \in M \cup N\),

1. \((0, 0)\) and all Nash equilibrium points belong to \(U_k\). Furthermore, their degree in the graph induced by \(U_k\) is exactly one.

2. The degree of every other vertex in the graph induced by \(U_k\) is two.
Proof. First, note that the label set of $(0,0)$ and any Nash equilibrium is exactly $M \cup N$, so $(0,0)$ and all Nash equilibrium points are in $U_k$ for any $k$. Furthermore, let $v = (v_1, v_2)$ be $(0,0)$ or any Nash equilibrium point. Without loss of generality, suppose $k \in L(v_1)$, where $v_1$ is a corner point of the polytope $P'$. Among all edges in $G_1$ that $v_1$ is incident to, there is only one direction leading to a vertex $v'_1$ without label $k$ (i.e. loosing the binding constraint corresponding to label $k$). It is easy to see that $(v'_1, v_2) \in U_k$, therefore there is only one neighbor of $v$ in $U_k$.

For part (2), let $v = (v_1, v_2)$ be any other point in $U_k$. Then there must be a duplicated label in $L(v_1)$ and $L(v_2)$, denoted by $l$. Similarly to (2), there is exactly one direction of $v_1$’s edges in $P'$ to drop the label $l$, and the new vertex $v'_1$ has all labels $v_1$ has except $l$, so $(v'_1, v_2) \in U_k$. It is symmetric for $v_2$. Hence there are two neighbors of $v$ in $U_k$. \hfill \Box

In other words, in a non-degenerate bimatrix game $(A,B)$ the set of k-almost completely labeled vertices in $G$ and their induced edges consist of disjoint paths and cycles. The endpoints of the paths are the artificial equilibrium $(0,0)$ and the equilibria of the game.

**Corollary 4.3.5.** A non-degenerate bimatrix game has an odd number of Nash equilibria.

For the symmetric evolutionary game, we assume $A = B$ and $x = y$. W.L.O.G., we also assume that all entries of $A$ are non-negative. Otherwise we can apply the Proposition 3.2.1, and $A$ has no all-zero columns or rows. We define a polytope, which is as same as the feasible region for a linear programming problem as below: denote $A^i$ as the $i$-th row of $A$, and $A_j$ as the $j$-th column of $A$. We have a polytope:

$$P = \{x \in \mathbb{R}^N \mid x_i \geq 0, \quad x^T A_j \leq 1\}. \quad (4.10)$$

Every Nash Equilibrium can be described as a corner point of $P$. We say that $x \in P$ has label $k \in N = \{1,2,\ldots,n\}$ if either $x_k = 0$, or $x^T A_k = 1$. We assume that $P$ is simple. A d-dimensional polytope is simple if every vertex meets exactly $d$ of the defining inequalities with equality. Otherwise, the game is degenerate. Fortunately, most games are non-degenerate. But even for degenerate games, we can get rid of the degeneracy by adding a small perturbation to the payoff matrix.
Theorem 4.3.6.  *(Specialized Lemke-Howson procedure for evolutionary game)*

\( x \) has all the labels in \( N \) if and only if normalized \( x \) is a Nash Equilibrium.

*Proof.* Suppose \( x \) has all the labels in \( N \). Let \( N_1 = \{ i | x_i = 0 \} \), \( N_2 = \{ j | x^T A_j = 1 \} \). Since the number of labels \( x \) having is no more than \( n \), we have \( N_1 \cup N_2 = N \). For each label \( i \in N \) in \( x \), the \( i \)-th row is not used in \( x \), and for each label \( j \in N \) in \( x \), the \( j \)-th column for the other player is the best response to \( x \). So this is the best response, and we have a Nash Equilibrium.

On the other hand, if \( x \) is a Nash Equilibrium, we have \( N \setminus \text{Supp}(x) \subset L \) for those rows not used in \( x \), and \( \text{Supp}(x) \subset L \) for those columns being best responses to \( x \). Since the game is non-degenerate, we have \( L(x) = (N \setminus \text{Supp}(x)) \cup \text{Supp}(x) = N \), which means \( x \) has all the labels in \( N \).

The specialized Lemke-Howson algorithm for finding Nash equilibria of evolutionary games is therefore summarized as following.

**Algorithm 4.3.7.** *(Specialized Lemke-Howson algorithm for evolutionary games)*

**Input:** A Non-degenerate payoff matrix \( A \).

**Output:** A set of \( n \) optimal strategies, i.e. Nash Equilibria. They are found by starting from \( n \) different labels respectively.

- **Step 1.** Let \( x = 0 \)
- **Step 2.** Choose some \( k = k_0 \in N \)
- **Step 3.** If \( k \in N \), drop label \( k \) from \( x \)
- **Step 4.** Let \( l \) be the label which is added by dropping label \( k \)
- **Step 5.** If \( l = k \), terminate and the normalized current \( x \) is a Nash Equilibrium of this game; if \( l \neq k \), then let \( k = l \) and go back to step 3 to repeat this process
4.4 Complementarity method

By the discussion of optimality conditions of evolutionary games in chapter 2, the definition of Nash equilibrium motivates the complementarity conditions. It leads to the complementarity method for solving the direct evolutionary game by solving a quadratic programming problem. From chapter 2, we have the following theorem.

**Theorem 4.4.1. (Restricted complementarity conditions of direct evolutionary game)**

For $\pi$ being the general payoff function on finite space of frequency $S_n$ as defined above, $x^* \in S_n$ is a Nash equilibrium if and only if, there exists a scalar $\lambda \in \mathbb{R}$ such that

$$x_i^*(\lambda - \pi(1, x^*)) = 0, \text{ and } x_i^* \geq 0, \lambda - \pi(1, x^*) \geq 0,$$

(4.11)

for $i = 1, 2 \cdots, n$. For evolutionary games with $\pi(x, x) = x^TAx$, $x^* \in S_n$ is a Nash equilibrium if (5.11) is held.

The problem of searching Nash equilibrium $x^*$ is therefore equivalent to solving the optimization problem

Minimize: $x^T(\lambda 1 - Hx) \downarrow 0$

Subject to: $1^Tx = 1$, $\lambda 1 - E(A, x) \geq 0$, $x \geq 0$.  

(4.12)

Finally, by letting

$$U_H = \begin{pmatrix} -H & 1 \\ 0 & 0 \end{pmatrix}, \quad U_A = \begin{pmatrix} -A & 1 \\ 0 & 0 \end{pmatrix}, \text{ and } y = \begin{pmatrix} x \\ \lambda \end{pmatrix},$$

the direct evolutionary game, i.e. searching Nash equilibrium is summarized as a quadratic programming problem

Minimize: $y^TU_Hy \downarrow 0$

Subject to: $\sum_{i=1}^{n-1} y_i = 1$, $(U_Hy)_i \lor (U_Ay)_i \geq 0$, 

$$0 \leq y_i \leq 1, \text{ for } 1 \leq i \leq n - 1.$$

(4.13)

Quadratic programming provides an efficient way to obtain a Nash equilibrium on $S_n$. The number of Nash equilibriums in high dimensional problem, however, might be large and
even uncountably many. We propose a Metropolis type scheme based on the restricted complementarity condition and a geometric sampling on the simplex for the purpose of exhausting the Nash equilibriums as much as possible to characterize the set of Nash equilibriums on $S_n$.

The sampling space is $(x, \lambda) \in S_n \times \mathbb{R}$ and we first investiage the feasible parameters space of $\lambda^*$ for the construction of the algorithm.

**Theorem 4.4.2.** *(Estimates on $\lambda^*$)* The optimal average payoff $\lambda^*$ of the evolutionary game defined by the payoff matrix $A$, has the following estimates

$$\lambda^* \in (\lambda_l, \lambda_u) = \left(\frac{\mu_{\min}}{n}, \mu_{\max}\right),$$

where $\mu_{\min}$ and $\mu_{\max}$ are the smallest and largest eigenvalues of $H = (A + A^T)/2$.

**Proof.** By the constraints imposed by the restricted complementarily condition, there is $x^T(\lambda 1 - x^T A x) \geq 0$ and $x^T 1 = 1$. Hence,

$$\lambda \geq x^T Ax$$

\forall x \in S_n. Using the conclusion for Rayleigh quotient, (4.15) leads to

$$\lambda \geq x^T A x$$

$$\lambda \geq x^T \left(\frac{A + A^T}{2}\right) x$$

$$\lambda \geq \mu_{\min} ||x||^2$$

where $\mu_{\min}$ is the smallest eigenvalues of $H$. Applying Jensen’s inequality on $||x||^2$, (4.16) implies

$$\lambda \geq \frac{\mu_{\min}}{n},$$

which is the lower bound for $\lambda$.

Furthermore, it is noticed that $\lambda^* = (x^*)^T A x^*$ if $(x^*, \lambda^*)$ is the pair of global minimum of quadratic programming problem satisfying the restricted complementarily conditions. Using the above notations, where $\mu_{\max}$ denotes the largest eigenvalue of $H$, we have

$$\lambda^* \leq \mu_{\max} ||x^*||^2 = \mu_{\max} \sum_{i=1}^{n} (x^*_i)^2 \leq \mu_{\max} \sum_{i=1}^{n} x^*_i = \mu_{\max}$$

as $x^* \in S_n$. We therefore obtain the assertion.
Motivated by the sampling procedure of Metropolis type, we sample points uniformly on subset of $S_n$ that the points are far away from the historical sampling points and all of the obtained Nash equilibriums, and accept the new sampling with probability following exponential distribution whose rate parameter should be proportional to the difference between $x^T A x \cdot 1 - A x$. In other word, the sampling space for $x$ at step $k$, $X_k$, is a quotient set of $S_n$ and collections of sets containing historical sampling points and Nash equilibriums, i.e. $X_k \subset X_{k-1}$. With new sampling point as initial point, we obtain Nash equilibrium by adapting quadratic programing on (4.13).

**Algorithm 4.4.3.** (Complementarily method for direct games)

**Input:** Payoff matrix $A$, and a tolerance radius $r$ with suggested value $0.15$ to $0.01$.

**Output:** A set of optimal strategies, i.e. Nash equilibria. The smaller tolerance radius given, the more complete Nash equilibrium set will be returned.

- Step 1. Examine all the vertices of $S_n$, $v_i$, by the restricted complementarity condition. Remove the sets $S_0^{-1} = \bigcup_{i=1}^{n} (B_r(v_i) \cap S_n) \cup \bigcup_{j \geq 1} (B_r(x^*_j) \cap S_n)$ for obtained Nash equilibriums $x^*$ if there is any, to get $S_n^{-1} = S_n \setminus S_0^{-1}$

- Step 2. Uniformly sample point from $S_n^{-1} \setminus \bigcup_{j \geq 1} (B_r(x^*_j) \cap S_n)$ with $k = 1, 2, \cdots, n, \cdots,$ and sample the average payoff $\lambda$ from parameter space $[\lambda_l, \lambda_u]$, accept the sample point with probability $e^{-||Ax-(x^T A x) \cdot 1||}$

- Step 3. Examine the accepted sample point $x_0$ using the restricted complementarity condition and remove the set $(B_r(x_0) \cup B_r(x^*)) \cap S_n^{-1}$ as step 1

- Step 4. Terminate if $\text{vol} \left( \bigcup (B_r(x^*) \cup B_r(x_0)) \cap S_0^{-1} \right) \geq \frac{\sqrt{n}}{(n-1)!}$ which is the volumn of hyper-tetrahedron in $\mathbb{R}^{n-1}$
CHAPTER 5. EVOLUTIONARY GAME DYNAMICS

5.1 Introduction

Game dynamics can be discrete or continuous, stochastic or deterministic. Continued from chapter 2, we deal with the ‘replicator dynamics’ in this chapter. It starts with a ‘folk theorem’ connecting the dynamics with Nash equilibria, which offers some results on a classification of its long-term behavior, and investigates the concept of an evolutionarily stable equilibrium. Among the recurrent questions are whether variants of the ‘folk theorem’ remain valid, and whether dominated strategies get eliminated. A central result on general ‘adjustment dynamics’ shows that every reasonable adaptation process will fail, for some games, to lead to a Nash equilibrium. We conclude by stressing the close links of evolutionary game dynamics with Nash’s original proofs of his equilibrium theorem. The emphasis on replicator dynamics does not mean to suggest that it is as important as all other dynamics together, but it serves conveniently for expository purposes and reflects some of the history of the subject.

5.2 Replicator dynamics

Consider a population consisting of $n$ types, and let $x_i$ be the frequency of type $i$. The state of the population is given by a vector $\mathbf{x} \in S_n$. Now assume that the $x_i$’s are differentiable functions of time $t$, which requires to assume that the population is infinitely large or the $x_i$’s are expected values for an ensemble of populations, and postulate a law of motion for $\mathbf{x}(t)$. If individuals meet randomly and then engage in a symmetric game with payoff matrix $A$, then $(A\mathbf{x})_i$ is the expected payoff for the individual of type $i$ and $\mathbf{x}^T A \mathbf{x}$ is the average payoff in the population state $\mathbf{x}$. Assume that the per capita rate of growth, i.e. the logarithmic
derivative \( \frac{d}{dt} \log(x_i) = \frac{1}{x_i} \frac{dx_i}{dt} \) is given by the difference between the payoff for type \( i \) and the average payoff in the population. This yields the replicator equation

\[
\frac{dx_i}{dt} = x_i((Ax)_i - x^T A x), \quad i = 1, \ldots, n.
\]  

(5.1)

The replicator equation describes a selection process such as population genetics and chemical networks: more successful strategies spread in the population.

Since the hyperplanes \( \sum_i x_i = 1 \) and \( x_i = 0 \) are invariant, it follows that the unit simplex \( S_n \) is invariant, and from now on we shall consider only the restriction of (5.1) to \( S_n \), the state space of the population. The boundary faces

\[
S_n(I) = \{ x \in S_n : x_i = 0 \quad \text{for all} \quad i \in I \},
\]  

(5.2)

where \( I \) is any non-trivial subset of \( N \), are also invariant under (5.1), and so is the interior, \( \text{int}(S_n) \), of the state space, where \( x_i > 0 \) for all \( i \). Two simple facts will be frequently used:

1. adding a constant \( c_j \) to all entries in the \( j \)-th column of \( A \) does not affect the replicator equation;

2. whenever the power product \( P = \prod_i x_i^{\alpha_i} \) is defined, its time-derivative satisfies

\[
\frac{dP}{dt} = P \sum_i \alpha_i ((Ax)_i - x^T A x).
\]  

(5.3)

For the long-term behavior of the dynamics, a rest point \( z \) is stable if for every neighborhood \( U(z) \) there exists a neighborhood \( V(z) \) such that \( x \in V \) implies \( x(t) \in U \) for all \( t \geq 0 \). The rest point \( z \) is said to be attracting if it has a neighborhood \( U \) such that \( x(t) \to z \) for \( t \to +\infty \) holds for all \( x \in U \). It is asymptotically stable if it is both stable and attracting, and globally stable if it is stable and \( x(t) \to z \) for \( t \to +\infty \) whenever \( x_i > 0 \) for all \( i \) with \( z_i > 0 \). One cannot request convergence for all \( x \in S_n \) since boundary faces are invariant. Similar definitions are used for a closed set of rest points, or a compact invariant set.

5.3 Stability conditions: Nash equilibrium, KKT point, and ESS

The rest points of the replicator equation are the points \( x \in S_n \) satisfying \( (Ax)_i = x^T A x \) for all \( i \in \text{supp}(x) \). Thus a rest point in \( \text{int}(S_n) \) is a solution of the system of linear equations
$(Ax)_1 = \cdots = (Ax)_n$, and the rest points in the interior of each subface $S_n(I)$ are obtained similarly. In particular, the corners $e_i$ of the state simplex are always rest points. There is a close connection between the rest points of the replicator equation and the Nash equilibria given by the symmetric game with payoff matrix $A$. By [16], we have:

1. if $z$ is a Nash equilibrium, then it is a rest point;

2. if $z$ is a strict Nash equilibrium, then it is asymptotically stable;

3. if the rest point $z$ is the limit of an interior orbit (an orbit $x(t)$ in int$(S_n)$) for $t \to +\infty$, then $z$ is a Nash equilibrium;

4. if the rest point $z$ is stable, then it is a Nash equilibrium.

This is sometimes referred to as the folk theorem of evolutionary game theory. The converse statements are not true. Every interior rest point is a Nash equilibrium. At a boundary rest point $z$, the difference $(Az)_i - z^T Az$ is an eigenvalue for the Jacobian of the replicator equation whose eigenvector is transversal to the face $z_i = 0$. Hence a rest point $z$ is a Nash equilibrium if and only if all its transversal eigenvalues are nonpositive. This yields the existence of Nash equilibria in terms of population dynamics. The equation

$$\frac{dx_i}{dt} = x_i((Ax)_i - x^T Ax - n\epsilon) + \epsilon, \quad i = 1, \ldots, n.$$  \hspace{1cm} (5.4)

is a perturbation of the replicator equation (5.1) with a small $\epsilon > 0$ representing a constant immigration term. This equation maintains the relation $\sum_i \frac{dx_i}{dt} = 0$ on $S_n$ and the flow on the boundary points into the interior of $S_n$. By a variant of Brouwer’s fixed point theorem, there exists at least one rest point $z(\epsilon)$ in int$(S_n)$, and

$$(Az(\epsilon))_i - z(\epsilon)^T Az(\epsilon) - n\epsilon = -\frac{\epsilon}{z_i(\epsilon)} < 0.$$ \hspace{1cm} (5.5)

Any accumulation point $z$ of $z(\epsilon)$ (for $\epsilon \to 0$) is a Nash equilibrium.

We are assuming that the ‘types’ in the population correspond to the pure strategies given by the basis vectors $e_i$ spanning the simplex $S_n$ in the analysis above. Suppose now the types may also correspond to mixed strategies $\mathbf{p}(i) \in S_n$, with $i = 1, \ldots, n'$, $n' \leq n$. The average
payoff for an individual of type $p(i)$ against an individual of type $p(j)$ is $u_{ij} = p(i)^T A p(j)$, and if $x \in S_n'$ describes the types' frequencies in the population, then the average strategy within the population is $p(x) = \sum x_i p(i)$. The induced replicator equation $\frac{dx_i}{dt} = x_i((U x)_i - x^T U x)$ can be written as

$$\frac{dx_i}{dt} = x_i((p(i) - p(x))^T A p(x)). \quad (5.6)$$

The best-known concept of evolutionary game theory is the evolutionary stability. If all members in the population use such an evolutionarily stable strategy, or ESS, then no ‘mutant’ minority using another strategy can invade. A strategy $q \in S_n$ is said to be evolutionarily stable if for every $p \in S_n$ with $p \neq q$, the induced replicator equation describing the dynamics of the population consisting of these two types only (the resident using $q$ and the invader using $p$) leads to the elimination of the invader as long as the initial frequency of this invader is sufficiently small, i.e. below some ‘invasion barrier’ $\epsilon(p)$. If $x$ is the frequency of the invader, we have:

$$\frac{dx}{dt} = x(1 - x)[x(p^T A p - q^T A p) - (1 - x)(q^T A q - p^T A q)] \quad (5.7)$$

and hence the rest point $x = 0$ is asymptotically stable if and only if the following conditions are satisfied:

1. $p^T A q \leq q^T A q \quad (5.8)$
2. $p^T A q = q^T A q$ and $p^T A p < q^T A p. \quad (5.9)$

The first condition means that $q$ is a Nash equilibrium: no invader does better than the resident against the resident. The second condition states that if the invader does as well as the resident against the resident, then it does less well than the resident against the invader. (Note that $x = 0$ may well be asymptotically stable, and hence $q$ is an ESS, if the replicator dynamics (5.6) is bistable: in this case, type $p$ can invade if it enters the population with a frequency which is sufficiently high, larger than the ‘invasion barrier’ $\epsilon(p)$. )
Theorem 5.3.1. [6] The strategy $q$ is an ESS if and only if $\Pi_i x^q_i$ is a strict local Lyapunov function for the replicator equation, or equivalently if and only if

$$q^T A p > p^T A p$$

(5.10)

for all $p \neq q$ in some neighborhood of $q$. If $q \in \text{int}(S_n)$, then (5.9) holds for all $p \in S_n$.

The function $V(x)$ is said to be a Lyapunov function if $\frac{dV(x)}{dt} \geq 0$ for all $x$, and strict if equality holds only when $x$ is a rest point. In particular, an ESS is an asymptotically stable rest point, and an interior ESS is globally stable. The converse does not hold in general. But we have the following theorem.

Theorem 5.3.2. [6] The strategy $q \in S_n$ is an ESS if and only if it is strongly stable.

Here, $q$ is said to be strongly stable if, whenever it is in the convex hull of $p(1), \ldots, p(n') \in S_n$, the strategy $p(x(t))$ converges to $q$, under (5.6), for all $x \in S_n'$ for which $p(x)$ is sufficiently close to $q$. The relation between evolutionary and dynamic stability is particularly simple for the class of partnership games, defined by $A = A^T$, for which the interests of both players coincide. For such games, $q$ is an ESS if and only if it is asymptotically stable for (5.1). This holds if and only if it is a strict local maximum of the average payoff $x^T A x$.

Many interesting games have no ESS. Often, it is useful to consider a generalization: a set $G \subseteq S_n$ is said to be an evolutionary stable set if for all $x_g \in G$ and all $x \in S_n$

$$x^T A x \leq x_g^T A x_g$$

(5.11)

holds, and if for all $x_g \in G$ and $x \in S_n \setminus G$ for which equality holds,

$$x^T A x < x_g^T A x.$$  (5.12)

A singleton set $G = \{x_g\}$ is an evolutionary stable set if and only if $x_g$ is an ESS. All elements of an evolutionary stable set $G$ are Nash equilibria which are neutrally stable in the sense that for $x, x_g \in G$ the equality $x^T A x = x_g^T A x$ holds whenever $x^T A x_g = x_g^T A x_g$. A set $G$ is an evolutionary stable set if and only if each $x_g \in G$ has a neighborhood $U$ such that $x^T A x \leq x_g^T A x$ with equality if and only if $x \in G$. If $G$ contains an $x' \in \text{int}(S_n)$, then $U$
can be chosen to be $S_n$. An evolutionary stable set $G$ is uninvadable in the sense that there exists an $\epsilon > 0$ such that any strategy $x_g \in G$ cannot be invaded by a minority of $x \notin G$ as long as its frequency is below $\epsilon$. Any strategy in an evolutionary stable set is stable, and any evolutionary stable set is asymptotically stable. If an evolutionary stable set $G$ contains a point $x_g$ in $\text{int}(S_n)$, then all orbits in the interior of $S_n$ converge to $G$.

Moreover, as we have seen in chapter 2, for the connections among the Nash equilibrium for evolutionary game (2.2), the KKT point for the constrained optimization problem (2.17), and the ESS. We have the following propositions.

**Proposition 5.3.3.** If $x$ is a KKT point, and $\frac{1}{2}(A^T - A)x \geq 0$, then $x$ must be a Nash equilibrium. If $x$ is a Nash equilibrium, and $\frac{1}{2}(A^T - A)x \leq 0$, then $x$ must be a KKT point.

**Proposition 5.3.4.** If $x$ is a strict global maximizer, and $\frac{1}{2}(A^T - A)x \geq 0$, then $x$ must be an ESS.

**Proposition 5.3.5.** If $x$ is a Nash equilibrium, and $\frac{1}{2}(A^T - A)x \leq 0$, then if $x$ is a minimizer or saddle point, $x$ is not an ESS.

There are important connections among the optimal strategies and evolutionary stable strategies of an evolutionary game (2.11) and the fixed points and asymptotically stable states of the corresponding system of replicator equations (2.14). In fact, there are also similar relationships among the optimal strategies and evolutionary stable strategies of a symmetric evolutionary game (9.1) and the KKT points and optimal solutions of the corresponding generalized knapsack problem (2.17). We discuss these relationships along with the fixed points and asymptotically stable states of the corresponding system of replicator equations system (2.9).

It is not easy to investigate the evolutionary stability of the optimal strategies for an evolutionary game directly based on its definition 2.2.2. Now based on the theorems in section 2.2, we consider the strategies $y$ in a small neighborhood $U$ of the optimal strategy $x^*$ and check if no $y \neq x^*$ prevails $x^*$ such that $y^T Ay \geq x^{*T} Ay$. It turns out that this condition is necessary and also sufficient:
Theorem 5.3.6. [16] An optimal strategy $x^* \in S$ for an evolutionary game (2.11) is evolutionarily stable if and only if there is a small neighborhood $U$ of $x^*$, $U \subset S$ such that
\[ y^T A y < x^* T A y, \quad \text{for all } y \in U, \quad y \neq x*. \] (5.13)

For a symmetric evolutionary game, we have $x^* T A y = y^* T A x$ since $A$ is symmetric. Then, $y^T A y < x^* T A x^*$ for all $y \in U, \quad y \neq x^*$ since $y^T A x^* \leq x^* T A x^*$ for all $y \in S$. This implies that if $x^*$ is an evolutionary stable strategy for a symmetric evolutionary game, it must be a strict local maximizer of the corresponding generalized knapsack problem: $\max\{ x^T A x : x \in S \}$. It turns out that the converse is also true.

Theorem 5.3.7. An optimal strategy $x^* \in S$ for a symmetric evolutionary game (9.1) is evolutionarily stable if and only if it is a strict local maximizer of the corresponding generalized knapsack problem (2.17).

Proof. Assume that an optimal strategy for $x^* \in S$ for a symmetric evolutionary game is evolutionarily stable. By Theorem 5.3.6, it is a strict local maximizer of the corresponding generalized knapsack problem (2.17). Now if $x^* \in S$ is a strict local maximizer of the corresponding generalized knapsack problem (2.17), there exists a neighborhood
\[ U = \{ x \in S : \| x - x^* \| < \epsilon \}, \]
such that for all $x \in U, \quad x \neq x^*$, it satisfies
\[ x^T A x < x^* T A x^*. \]

Now choose another neighborhood of $x^*$,
\[ U' = \{ y \in S : y = \frac{x + x^*}{2}, x \in U \}. \]

It implies $x = 2y - x^*$ and then
\[ (2y - x^*)^T A (2y - x^*) < x^* T A x^*, \quad \text{for all } y \in U', \quad y \neq x*. \]

It follows immediately that
\[ y^T A y < x^* T A y, \quad \text{for all } y \in U, \quad y \neq x*. \]

Therefore, $x^*$ is evolutionarily stable. \qed
Combine Theorem 5.3.7 with the necessary and sufficient conditions for the optimal solution to the generalized knapsack problem in (2.17) derived in section 2.2, we now have a set of necessary and sufficient conditions for the evolutionary stable strategy for the symmetric evolutionary game in (9.1).

**Theorem 5.3.8.** Let \( x^* \in S \) be an evolutionarily stable strategy for the symmetric evolutionary game in (9.1). Let the row rank of \( C'(x^*) \) be equal to \( m \). Let \( Z \in \mathbb{R}^{n \times (n-m)} \) be the null space matrix of \( C'(x^*) \). Then, \( Z^T AZ \) must be negative semi-definite.

**Theorem 5.3.9.** Let \( x^* \in S \) be an optimal strategy for the symmetric evolutionary game in (9.1). Let the row rank of \( C'(x^*) \) be equal to \( m \). Let \( Z \in \mathbb{R}^{n \times (n-m)} \) be the null space matrix of \( C'(x^*) \). If \( Z^T AZ \) is negative definite, then \( x^* \) must be an evolutionarily stable strategy.

Consider a solution \( x(t) \) to the replicator equation system in (2.9), \( x(0) = x_0 \). It is well-known that \( V(x(t)) = \frac{x(t)^T Ax(t)}{2} \) can be used as a Lyapunov function for the convergence of \( x(t) \) [51; 16]. By differentiating \( V \) with respect to \( t \), we obtain

\[
V_t'(x(t)) = \sum_i x_i(t) |x(t)^T A x(t)(A x(t))_i|^2 \geq 0 \tag{5.14}
\]

with the equality holds only when \( x(t) \) is equal to an equilibrium solution \( x^* \) of (2.9). Therefore, \( x(t) \) converges to \( x^* \) asymptotically, and so do all the solutions started within a small neighborhood \( U \) of \( x^* \), where \( V_t'(x(t)) > 0 \) for all \( x(t) \neq x^* \). As these solutions \( x(t) \) converge to \( x^* \), \( V(x(t)) \) increases monotonically to a maximum \( V(x^*) = \frac{x^*^T A x^*}{2} \). Since \( V(x) < V(x^*) \) for all \( x \in U, x \neq x^* \), \( x^* \) is a strict local maximizer of \( V \). It follows that \( x^* \) must be a strict local maximizer of the corresponding generalized knapsack problem (2.17) and hence an evolutionary stable strategy of the corresponding symmetric evolutionary game (9.1) by Theorem 5.3.7. Since an evolutionarily stable strategy for an evolutionary game is always an asymptotically stable equilibrium point of the corresponding system of replicator equations by Theorem 2.2.3, we then obtain the following theorem.

**Theorem 5.3.10.** [51] A fixed point \( x^* \in S \) of a replicator equation system (2.9) is asymptotically stable if and only if it is an evolutionary stable strategy for the corresponding symmetric
By all above, we have that the optimal strategies of the symmetric evolutionary game in (9.1) are the KKT points of the corresponding generalized knapsack problem in (2.17); and the evolutionary stable strategies of the symmetric evolutionary game in (9.1) are the strict local maximizers of the corresponding generalized knapsack problem in (2.17) and the asymptotically stable equilibrium points of the corresponding replicator equation system in (2.9). Now we summarize the relationships among all these different types of solutions in the following theorem.

**Theorem 5.3.11.** Let a symmetric evolutionary game be given in (9.1) and a corresponding generalized knapsack problem in (2.17) and a corresponding replicator equation system in (2.9). Let \( O \) and \( E \) be the sets of optimal strategies and evolutionary stable strategies for the symmetric evolutionary game in (9.1), respectively. Let \( K \) and \( S \) be the sets of KKT points and strict local maximizers of the generalized knapsack problem in (2.17), respectively. Let \( F \) and \( A \) be the sets of fixed points and asymptotically stable equilibrium points of the replicator equation system in (2.9). Then, we have

\[
F \supset O = K \supset A = S = E.
\]  

(5.15)
CHAPTER 6. SPECIAL TOPIC: SPARSE AND DENSE NASH EQUILIBRIA

6.1 Introduction

Given an evolutionary game defined by a matrix $A$. As we have discussed in chapter 2 and 4, the problem of solving Nash equilibrium can be formulated as following

$$\min_{x \in \mathbb{R}^n} : x^T (\lambda 1 - Ax) \to 0$$
subject to : $$\sum_{i=1}^{n} x_i = 1, \quad x_i \geq 0, \quad \lambda - (Ax)_i \geq 0.$$  \hfill (6.1)

The sparsity of the Nash equilibria has not been studied. We consider the following optimization problem

$$\min_{x \in S_n} : \|x\|_0$$
subject to : $$x^T (\lambda 1 - Ax) = 0, \quad \sum_{i=1}^{n} x_i = 1,$$ $$x_i \geq 0, \quad \lambda - (Ax)_i \geq 0,$$ \hfill (6.2)

where $\|x\|_0$ stands for the number of the nonzero components of $x$. The solution of the above problem is called the sparse solution of the evolutionary game defined by $A$.

It is clear that the problem (6.2) can be solved by the specialized Snow-Shapley method derived from section 4.1. However, there are always $2^n$ possible submatrices for calculating the complete set of Nash equilibria. Especially, for each submatrix $M$ with dimension $m$, $1 \leq m \leq n$, $\text{adj}(M)$ must be calculated. It involves the computation of $A^{-1}$ and $\det(A)$, both with complexity of $O(m^3)$.

The concept of Nash equilibrium is widely applied in economics, ecology, genetic selection, and political science. The possible applications of the sparse or dense solution can also
be significant in these fields. The sparsity of Nash equilibria can imply the diversity of components in a community consists of multiple species. The set of the densest Nash equilibria in such a game provides a state of co-existence of species with highest diversity. In financial market, the Nash equilibrium with higher density can imply the investment strategy that produces the same payoff as other optimal solutions but with the money distributed in a more diversified way.

### 6.2 Dual method for direct evolutionary games

The complementarity method gives multiple Nash equilibria for a given system by adjusting parameters. However, it cannot guarantee the completeness. Lemke-Howson method also has this flaw. Snow-Shapley method is the only one that can always find the complete solution set. But huge computation would be involved when the community is of high dimensions.

To lower the cost, we try to solve a so-called dual problem from the original optimization problem. With less computation cost, it returns the complete set of Nash equilibrium. To do this, consider the optimization problem we got from complementarity method:

$$
\min_{x \in \mathbb{R}^n} : \ x^T(\lambda \mathbf{1} - Ax) \to 0
$$

subject to : $\sum_{i=1}^{n} x_i = 1, \ x_i \geq 0, \ \lambda - (Ax)_i \geq 0.$ (6.3)

Instead of solving the vector of species $x$ directly, if $A$ is non-singular, consider

$$
y = \lambda \mathbf{1} - Ax,
$$

then we have

$$
x = \lambda A^{-1} \mathbf{1} - A^{-1} y,
$$

where $\mathbf{1}$ is an n-dimension vector of ones($\mathbf{1} = (1, \cdots, 1)^T$). The same problem can be represented by $y$ as following

$$
\min_{y \in \mathbb{R}^n} : \ (\lambda A^{-1} \mathbf{1} - A^{-1} y)^T y = y^T(\lambda A^{-1} \mathbf{1} - A^{-1} y) \to 0
$$

subject to : $\mathbf{1}^T(\lambda A^{-1} \mathbf{1} - A^{-1} y) = 1, \ (\lambda A^{-1} \mathbf{1} - A^{-1} y)_i \geq 0, \ y_i \geq 0.$ (6.6)
The complementarity condition tells either \( y_i = 0 \), or \( x_i = (\lambda A^{-1} \mathbf{1} - A^{-1}y)_i = 0 \) for all \( 1 \leq i \leq n \). Notice that the first constraint
\[
\mathbf{1}^T(\lambda A^{-1} \mathbf{1} - A^{-1}y) = 1
\] (6.7)
can be written as
\[
c^T(\lambda A^{-1} \mathbf{e} - A^{-1}y) = 1,
\] (6.8)
where \( c_i = 1 \) if \( y_i = 0 \), and \( c_i = 0 \) if \( x_i = 0 \). It implies
\[
\lambda = \frac{1 + c^T A^{-1}y}{c^T A^{-1} \mathbf{1}}.
\] (6.9)
Now consider \( y_i \neq 0 \), \( (\lambda A^{-1} \mathbf{1} - A^{-1}y)_i = 0 \) for some \( i \). We need
\[
\left( \frac{(1 + c^T A^{-1}y)A^{-1}}{c^T A^{-1} \mathbf{1}} - A^{-1}y \right)_i = 0.
\] (6.10)
It is equivalent to
\[
[(c^T A^{-1} \mathbf{1}A^{-1} - A^{-1}1c^T A^{-1})y]_i = (A^{-1} \mathbf{1})_i.
\] (6.11)
A linear system \( B \mathbf{y} = \mathbf{p} \) can be obtained by setting
\[
B = c^T A^{-1} \mathbf{1}A^{-1} - A^{-1}1c^T A^{-1}, \quad \mathbf{p} = A^{-1} \mathbf{1},
\] (6.12)
where \( b_{jk} = b_{kj} = 0 \) for all \( 1 \leq k \leq n \), \( b_{jj} = 1 \), and \( p_j = 0 \) if \( y_j = 0 \). Solving this linear system gives a potential Nash equilibrium. The complementarity condition implies that the total number of possible Nash equilibria is \( 2^n \). The theorem for dual method is summarized as following.

**Theorem 6.2.1.** (Dual method for evolutionary games)  
*Given a evolutionary game defined by a non-singular matrix \( A \in \mathbb{R}^{n \times n} \), \( x^* \in S_n \) is a Nash equilibrium if and only if there is a matrix
\[
B = c^T A^{-1} \mathbf{1}A^{-1} - A^{-1}1c^T A^{-1},
\] (6.13)
and a vector
\[
\mathbf{p} = A^{-1} \mathbf{1},
\] (6.14)*
where \( \mathbf{1} = (1, \cdots, 1)^T \), \( c_i = 0 \) if \( x_i = 0 \), such that

\[
B \mathbf{y}^* = \mathbf{p}, \tag{6.15}
\]

where

\[
\mathbf{x}^* = \lambda \mathbf{A}^{-1} \mathbf{1} - \mathbf{A}^{-1} \mathbf{y}^*, \quad \lambda = \frac{1 + \mathbf{c}^T \mathbf{A}^{-1} \mathbf{y}^*}{\mathbf{c}^T \mathbf{A}^{-1} \mathbf{1}}. \tag{6.16}
\]

Based on Theorem 6.2.1, the algorithm is summarized as following.

**Algorithm 6.2.2.** (Dual method for evolutionary game)

**Input:** Payoff matrix \( \mathbf{A} \).

**Output:** Complete set of optimal strategies, i.e. Nash equilibria.

- Step 1. Examine the matrix \( \mathbf{A} \), which must be non-singular.

- Step 2. Generate the indices \( i_1, \ldots, i_m \) where \( m < n \) and assume \( x_{ij} = 0 \) for all \( 1 \leq j \leq m \). Compute \( \mathbf{y}^* \) using (6.14), and discard the solution if \( \mathbf{y}^* \) has any negative component.

- Step 3. Compute the solution \( \mathbf{x}^* \) by (6.16). Discard the solution if it violates the necessary and sufficient conditions in Theorem 2.3.1.

- Step 4. All the legitimate solution will form the set of Nash equilibriums of evolutionary game with payoff matrix \( \mathbf{A} \).

To test the performance of this dual method, we applied both specialized Snow-Shapley method and dual method to a group of 10 games, which are defined by 10 randomly generated \( 20 \times 20 \) payoff matrices. For each game, we search for the set of sparsest Nash equilibria, the set of densest, and the complete set of all optimal solutions. The time costs are based on the average of five experiments. The results of using the specialized Snow-Shapley method are summarized in Table 6.1.
It is obvious that the computation for the set of sparsest Nash equilibria is efficient. However, we expect better results on the densest and complete set. Now we look at the performance of the dual method for these two sets in Table 6.2. On average, the dual method needs 119s for completing the computation of all possible Nash equilibria while the specialized Snow-Shapley method needs 724s; the dual method needs 61s for the densest Nash equilibria while the specialized Snow-Shapley method needs 451s.

Table 6.2 Computation time (in seconds) for direct game by dual method

<table>
<thead>
<tr>
<th>Game No.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete</td>
<td>120</td>
<td>118</td>
<td>119</td>
<td>119</td>
<td>119</td>
<td>118</td>
<td>118</td>
<td>119</td>
<td>119</td>
<td>119</td>
</tr>
<tr>
<td>Densest</td>
<td>114</td>
<td>48</td>
<td>86</td>
<td>67</td>
<td>15</td>
<td>7</td>
<td>86</td>
<td>108</td>
<td>68</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 6.1 Computation time (in seconds) for direct game by specialized Snow-Shapley method

<table>
<thead>
<tr>
<th>Game No.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete</td>
<td>692</td>
<td>790</td>
<td>693</td>
<td>711</td>
<td>711</td>
<td>788</td>
<td>700</td>
<td>686</td>
<td>738</td>
<td>733</td>
</tr>
<tr>
<td>Sparsest</td>
<td>1.83</td>
<td>0.02</td>
<td>0.43</td>
<td>0.41</td>
<td>0.06</td>
<td>0.02</td>
<td>0.02</td>
<td>0.44</td>
<td>0.02</td>
<td></td>
</tr>
<tr>
<td>Densest</td>
<td>678</td>
<td>426</td>
<td>605</td>
<td>531</td>
<td>186</td>
<td>94</td>
<td>603</td>
<td>668</td>
<td>531</td>
<td>185</td>
</tr>
</tbody>
</table>
CHAPTER 7. NUMERICAL EXAMPLES

In this chapter, the algorithms for direct evolutionary game are demonstrated in the examples of gene mutations for malaria resistance and plant succession problems from [47].

7.1 Gene mutations for malaria resistance

A genetic selection problem and in particular, the problem for allele selection at a single or multiple genetic loci can be formulated as a symmetric evolutionary game [3]. Recall that the fitness of different allele pairs or in other words, different genotypes at a given genetic locus can be given in a matrix with the rows corresponding to the choices for the first allele and the columns to the choices for the second allele. If there are \( n \) different alleles, there will be \( n \) different choices for both the first and second alleles, and the fitness matrix will be an \( n \times n \) matrix. With such a fitness matrix, the allele selection game can be defined with the choices of the first and second alleles as the strategies for player I and player II of the game, where player I can be considered as a specific individual and player II as a typical individual in the given population. If there are \( n \) different alleles, the strategy for player I can be represented by an \( n \)-dimensional vector \( \mathbf{x}, \mathbf{x} \geq 0, \sum_i x_i = 1 \), and the strategy for player II by an \( n \)-dimensional vector \( \mathbf{y}, \mathbf{y} \geq 0, \sum_i y_i = 1 \). Let the fitness matrix be given by \( A \in \mathbb{R}^{n \times n} \). Let \( S_n = \{ \mathbf{x} \in \mathbb{R}^n : \mathbf{x} \geq 0, \sum_i x_i = 1 \} \). The average fitness of an allele choice \( x \in S_n \) in an allele population \( y \in S_n \) will be \( \mathbf{x}^T A \mathbf{y} \). We then want to find an optimal choice of \( x^* \in S_n \) such that

\[
\mathbf{x}^* A \mathbf{x}^* \geq \mathbf{x}^T A \mathbf{x} \quad \text{for all} \quad \mathbf{x} \in S_n,
\]

i.e., in allele population \( \mathbf{x}^* \), any individual with allele choice \( \mathbf{x} \) other than \( \mathbf{x}^* \) will not have a better average fitness than allele choice \( \mathbf{x}^* \). Note that the fitness for allele pair \((i, j)\) usually
is the same as that for \((j, i)\). Therefore, the fitness matrix for allele selection is typically symmetric, and the game in (7.1) is then a symmetric evolutionary game.

The dynamics of the selection game in (7.1) can be described by using a system of replicator equations. Let \(x\) be a state of the allele population at time \(t\) or in other words, at generation \(t\), with \(x_i\) being the portion of allele \(i\) in the population. Assume \(x\) is a continuous function of \(t\). Then, the increasing rate of \(x_i\) will be proportional to the average fitness gain \((Ax)_i\) of allele \(i\) over the average fitness \(x^T A x\) of the population, i.e.,

\[
x'_i = x_i((Ax)_i - x^T A x), \quad x_i \geq 0, \quad \sum_i x_i = 1, \quad i = 1, \ldots, n.
\] (7.2)

The selection game in (7.1) is a symmetric evolutionary game. The dynamic system in (7.2) is a replicator equation system. By Theorem 5.3.11, an optimal solution \(x^*\) of selection game (7.1) is necessarily a fixed point of system (7.2), when \((x^*_i)' = 0\) for all \(i\), and the population has reached equilibrium. If \(x^*\) is also evolutionarily stable, there will be a solution \(x(t)\) of system (7.2) asymptotically converging to \(x^*\) as \(t\) goes to infinity, and vice versa. Here, a solution \(x(t)\) of system (7.2) shows not only what but also how the allele population may evolve to over time. As we have discussed in previous sections, the selection game in (7.1) and the associated system (7.2) can be studied with a generalized knapsack problem:

\[
\max_{x \in \mathbb{R}^n} \frac{x^T A x}{2}
\]

subject to:

\[
\sum_i x_i = 1, x \geq 0.
\] (7.3)

By Theorem 5.3.11, an optimal solution \(x^*\) of (7.1) is equivalent to a KKT point of the generalized knapsack problem in (7.3), and if it is evolutionarily stable, it must correspond to a strict local maximizer of (7.3), and vice versa. In addition, the optimality and stability conditions derived in previous sections all apply to the selection game in (7.1). We demonstrate the applications of these results with a game related to gene mutations for malaria resistance.

We can find all the solutions for a symmetric evolutionary game by using the necessary and sufficient conditions in Theorem 2.3.2. To verify the stability, all we need is to examine the solutions to see if they are strict local maximizers of the corresponding generalized knapsack
problem, which is more straightforward and easier to check than conventional approaches that
require more theoretical proofs.

Now we look at a case related to the gene mutations for malaria resistance. In Africa
and Southeast Asia, where human population has been exposed to serious malaria infection,
certain genetic mutations have survived for a gene that codes the hemoglobin proteins of
blood cells. These mutations resist malaria infections, but may cause other serious illness
as well when in homozygote forms such as the sickle cell disease. Here we consider three
well-studied allele forms of this gene, the wild type, S-mutation, and C-mutation, denoted by
W, S, and C alleles [44]. The normal genotype would be WW, but subnormal ones include
WS, WC, and SC, which may have malaria resistance functions. Other forms, SS and CC,
may cause other illness. These functions can be described with a 3 by 3 fitness matrix A,
with rows corresponding to the choices of W, S, and C for the first allele, and the columns
to the choices of W, S, and C for the second allele, when forming the allele pairs or in other
words, the genotypes. Based on recent study on malaria infection [44], this fitness matrix is
estimated as follows:

\[
A = \begin{pmatrix}
0.89 & 1.00 & 0.89 \\
1.00 & 0.20 & 0.70 \\
0.89 & 0.70 & 1.31
\end{pmatrix}
\]

From this matrix, we see that the genotype WS has good fitness, while CC is the best.
The genotype WW is not very good because it is susceptible to malaria infection, while
SS is the worse because it causes the sickle cell disease. We may wonder how the alleles
will eventually distribute in the population under such selection pressure. We have solved a
symmetric evolutionary game with this fitness matrix and obtained three solutions:

\[
x_1^* = \begin{pmatrix}
0 \\
0 \\
1
\end{pmatrix}, \quad x_2^* = \begin{pmatrix}
0.879 \\
0.121 \\
0
\end{pmatrix}, \quad x_3^* = \begin{pmatrix}
0.832 \\
0.098 \\
0.070
\end{pmatrix}.
\]

The first solution suggests that the population may end up with all C alleles since the
genotype CC seems have the best fitness. The second solution suggests a large portion of W
alleles, with a small percentage of S alleles, which increases the resistance to malaria infection,
yet does not have a large chance for $SS$ combinations. The third solution means that the three alleles may co-exist.

We have also solved a corresponding generalized knapsack problem and replicator equation system with the above matrix $A$, using our toolbox in Matlab (TEDA). It turned out that we have only found two local maximizers for the generalized knapsack problem corresponding to $x_1^*$ and $x_2^*$. At least, computationally, we have not found $x_3^*$ as a local maximizer. In our replicator equation system solutions, no matter what initial conditions are, we all end up with solutions converging to either $x_1^*$ or $x_2^*$, but not $x_3^*$. These results show computationally that $x_1^*$ and $x_2^*$ are evolutionarily stable, while $x_3^*$ may not be a stable solution. Indeed, at solution $x_3^*$, the only active constraint for the generalized knapsack problem is $\sum_i x_i = 1$. The null space matrix $Z$ for the Jacobian of this equation can be constructed as

$$Z = \begin{pmatrix} 1 & 0 \\ -1 & 1 \\ 0 & -1 \end{pmatrix}.$$  

We then have the reduced Hessian of the generalized knapsack problem to be

$$Z^T A Z = \begin{pmatrix} -0.91 & 0.61 \\ 0.61 & 0.11 \end{pmatrix},$$

and the eigenvalues of this reduced Hessian are $-1.1951$ and $0.3951$. By Theorem 2.3.9, $x_3^*$ is not a local maximizer of the generalized knapsack problem and hence by Theorem 5.3.7, it is not evolutionarily stable.

Based on the above analysis, we would predict that $x_3^*$ for the co-existing of three alleles in the population will never happen because it is unstable. The solution $x_1^*$ corresponds to a global maximizer of the generalized knapsack problem. Based on our simulation, it also has a large attraction region in the sense that most solutions will converge to $x_1^*$ unless the initial value for $C$ allele is very small, say less than 5%. In current population, $C$ allele is indeed rare and therefore, the population does not have much chance to evolve to this state. The population have typically a large percentage of $W$ alleles, a small percentage of $S$ alleles,
and some rare $C$ alleles, and therefore, $x^*_2$ could be the most likely and stable state of the population in the end.

### 7.2 An invading model in vegetation dynamics

A plant succession problem can also be formulated as an evolutionary game [47]. A plant community can be represented by a vector $x$ containing the proportional cover, $x_i$, of each of its species, with open ground being treated as another species. Consider the invading species that have been present for a long time in the form of seeds, rhizomes or suppressed individuals with the dominant species in the community preventing their increase. The new species may increase only after a change in community composition. An invader depends on the identity and abundance of the species already present, and it is similar that the pay-off to a new player depends on the strategies already established in game theory.

The fitness of different species competing in a community can be given in a payoff matrix. The rows corresponds to the invading species and the columns to the species in the community. Let $A$ be a payoff matrix, where $a_{ij}$ indicates the proportional payoff to species $i$ if surrounded by species $j$. Here ‘payoff’ is to be interpreted as per unit cover rate of increase of the cover of $i$. We assume that offspring play the same strategies as their parents, and that each individual plays a fixed pure strategy. If there are $n$ different species, there will be $n$ different choices for both the invading and defending species, and the fitness matrix will be an $n \times n$ matrix.

The plant succession game can be defined with the choices of the invading and existing species as the strategies for player I and player II of the game, where player I can be considered as a specific individual and player II as a typical individual in the given population. If there are $n$ different species, the strategy for player I can be represented by an $n$-dimensional vector $x$, $x \geq 0$, $\sum_i x_i = 1$, and the strategy for player II by an $n$-dimensional vector $y$, $y \geq 0$, $\sum_i y_i = 1$. Let $S_n = \{x \in \mathbb{R}^n : x \geq 0, \sum_i x_i = 1\}$. We can then compute the pay-off to species $i$ within a community $x$ as

$$\sum_j a_{ij}x_i = (Ax)_i.$$  (7.4)
So the average pay-off to the species of community $x$ is

$$\sum_i (Ax)_i = x^T Ax.$$  \hfill (7.5)

Moreover, we have the average pay-off to the species of community $x$ with invading community $y$ is $x^T Ay$. If the rate of growth of species $i$ is proportional to its payoff value, we have the following replicator equation system

$$\frac{dx_i}{dt} = x_i[(Ax)_i - x^T Ax].$$  \hfill (7.6)

If a community $x$ cannot be invaded successfully by any one or more of the available species not in $x$, we say $x$ is a non-invadable community. It corresponds to an evolutionary stable strategy. A community $e$ will be non-invadable if for all $x \in S_n \setminus \{e\}$ either

$$x^T Ae < e^T Ae,$$  \hfill (7.7)

where $e$ beats all other communities, or

$$x^T Ae = e^T Ae, \quad \text{and} \quad x^T Ax < e^T Ax,$$  \hfill (7.8)

where $x$ and $e$ are matched but $e$ does better in $x$ than $x$ itself.

Now we apply this game theoretic model to succession data. Data set from Sousa [42] has 6 species involved and the open space is treated as the 7-th state. It is about a community of macrophytic algae on the blocks, which were set out near Santa Barbara, California, and their percentage cover was scored over 2.5 years. The invasion matrix is estimated as in Table 7.1.
Table 7.1 Payoff matrix from [47] regarding plant successions problem

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.Open</td>
<td>0.00</td>
<td>-0.22</td>
<td>0.04</td>
<td>0.15</td>
<td>0.44</td>
<td>-0.82</td>
<td>0.64</td>
</tr>
<tr>
<td>2.Ulva spp.</td>
<td>0.21</td>
<td>0.00</td>
<td>-0.18</td>
<td>-0.05</td>
<td>-0.69</td>
<td>0.71</td>
<td>-0.26</td>
</tr>
<tr>
<td>3.Gigartina canaliculata</td>
<td>0.30</td>
<td>0.54</td>
<td>0.00</td>
<td>-0.41</td>
<td>-0.10</td>
<td>1.23</td>
<td>-0.54</td>
</tr>
<tr>
<td>4.Gigartina leptorynchos</td>
<td>0.38</td>
<td>-0.22</td>
<td>0.90</td>
<td>0.00</td>
<td>-1.61</td>
<td>0.62</td>
<td>0.53</td>
</tr>
<tr>
<td>5.Gelidium coulteri</td>
<td>-0.29</td>
<td>-0.22</td>
<td>-0.37</td>
<td>0.40</td>
<td>0.00</td>
<td>-1.18</td>
<td>-0.10</td>
</tr>
<tr>
<td>6.Rhodoglossum affine</td>
<td>2.07</td>
<td>2.05</td>
<td>2.33</td>
<td>2.57</td>
<td>2.77</td>
<td>0.00</td>
<td>1.58</td>
</tr>
<tr>
<td>7.Chthamalus fissus</td>
<td>0.71</td>
<td>0.94</td>
<td>0.65</td>
<td>0.37</td>
<td>0.05</td>
<td>0.51</td>
<td>0.00</td>
</tr>
</tbody>
</table>

By using specialized Snow-Shapley method, specialized Lemke-Howson method, complementarity method, and dual method, the only unintradable community is one with Gigartina canaliculata covering nearly 35% and Rhodoglossum affine covering 65%. In Sousa’s observation [42], these two species were often dominating the community.
CHAPTER 8. SOLVER IMPLEMENTATION: TEDA

In this chapter, we introduce a toolbox written in Matlab (Toolbox for Evolutionary Dynamics Analysis, TEDA) for analysis of evolutionary games. Based on the theoretical results in previous chapters, TEDA provides the solver for both inverse and direct games, stability analysis, and replicator dynamics simulation. Solving inverse evolutionary game is demonstrated based on the data regarding Barnes’ problem [48], and the algorithms for direct evolutionary game are elaborated based on the estimated payoff matrix for a plant succession problems from [47].

8.1 Toolbox for evolutionary dynamics analysis (TEDA)

TEDA is a toolbox based on GUI environment in Matlab. With all necessary algorithms incorporated, TEDA consists of four modulus: inferences of the payoff matrix from experimental data, searching Nash equilibria, stability investigation, and simulation of the replicator dynamics. The main interface of TEDA is shown in Figure 8.1.
One can input data into TEDA by directly typing numbers or upload excel files locally. TEDA will demonstrate the output in the toolbox windows or separate windows if necessary and also export excel files containing results into the folder where TEDA is executed. For the inverse evolutionary game, TEDA will export ps files containing figures of histograms of the estimated components of payoff matrix and 95% credible bands of data into the same folder. Dynamical simulation will be export into a separated pdf file by TEDA as well. Selection of method and input of initial values or algorithm parameters are executed by direct GUI actions which are demonstrated below.

8.2 Example of inverse evolutionary game

The Barnes’ problem [48] is originally used to describe chemical reactions, and it is in the form of Lotka-Volterra equation. In [48], the authors provide data of predator-prey type for two species which is in the Table 8.1. Considering the data set being modeled by evolutionary game modeling, we demonstrate how our toolbox performs as below.

Figure 8.1 Main interface of TEDA.
Table 8.1 Data from [48] regarding Barnes’ problem

<table>
<thead>
<tr>
<th>t</th>
<th>0</th>
<th>0.5</th>
<th>1</th>
<th>1.5</th>
<th>2</th>
<th>2.5</th>
<th>3</th>
<th>3.5</th>
<th>4</th>
<th>4.5</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>y1</td>
<td>1.0</td>
<td>1.1</td>
<td>1.3</td>
<td>1.1</td>
<td>0.9</td>
<td>0.7</td>
<td>0.5</td>
<td>0.6</td>
<td>0.7</td>
<td>0.8</td>
<td>1.0</td>
</tr>
<tr>
<td>y2</td>
<td>0.3</td>
<td>0.35</td>
<td>0.4</td>
<td>0.5</td>
<td>0.5</td>
<td>0.4</td>
<td>0.3</td>
<td>0.25</td>
<td>0.25</td>
<td>0.3</td>
<td>0.35</td>
</tr>
</tbody>
</table>

In the parameter estimate (inverse game) part of TEDA, we start by choosing the data file 'Barne.xls'. The format of the data should be a \((d+1) \times n\) matrix where \(d\) is the number of species in the data set, \(n\) is the number of the measurement time point. The first row of the data is the time point (nonnegative non-repeated real numbers), the 2-nd to \((d+1)\)-th rows are the data/responses.

The input parameter variance of the data is used for the purpose of producing 95% credible bands of data and is suggested to be chosen between 0.05 and 5. Smoothing parameter is important to reduce the MRSE of the estimation and the suggested value is between 0 to 0.01. Variance and smoothing parameters can be adjusted based on the output credible bands of the data to improve the estimation and inference results. Number of iteration is used to produce inferences of payoff matrix via parametric bootstrap and is suggested to be at least 1000.

Figure 8.2 Parameter estimation for Barnes’ Problem
After uploading data and inputting parameters, the inverse game is executed by clicking the button of ‘Process’.

Three excel files named payoff.xls, tempmatrix.xls and tempvec.xls will be generated in the same folder. The file payoff.xls contains only the estimated payoff matrix of interests. File tempmatrix.xls records all the important statistical output in the matrix form and tempvec.xls stores them in the vector form. The output quantities include estimated payoff matrix, estimated components of the payoff matrix in vector form, standard deviation of the estimates MRSE the root mean squares of the estimates for the components of payoff matrix, MAE the Mean average error of the estimates for the components of payoff matrix, ci95t the 95% credible interval of the estimates for the components of payoff matrix, ci95tb the 95% basic bootstrap interval of the estimates for the components of payoff matrix, estimated mean for the payoff matrix, estimated bias of estimates for the components of payoff matrix, and the estimated covariance based on bootstrap of the payoff components.
Figure 8.4 Inference statistics of the estimated payoff matrix

All the plots about 95% credible bands with data will be saved as 95CBOFdata.ps, and can be used directly in latex or converted into .pdf. All the plots about the estimated marginal distribution (in the form of histograms) of payoff matrix components will be saved as MDofPayoof.ps.

![Figure 8.4](image1)

As plotted in Figure 8.5, 95% credible bands with data together are saved as 95CBOF-data.ps. They provide a criterion to adjust the hypothetical variance and smoothing param-

![Figure 8.5](image2)
eters to improve the estimation. If the credible bands do not cover the estimated mean well or too broad, then one may choose smaller variance or large smoothing parameter.

### 8.3 Example of direct evolutionary game

To demonstrate the algorithms implemented within TEDA for the direct evolutionary games, the plant succession problem from [47] is used as a numerical example. As we have seen in chapter 7, the plant succession is modeled by replicator dynamics (7.6) with payoff matrix $A$. A plant community is represented by a vector $x$ where $x_i$ is the proportional cover of each of the species in this community. The open ground is considered as another species. The invasion matrix (payoff matrix) $A$ is estimated as in Table 8.2 from [47].

| Table 8.2 Payoff matrix from [47] regarding plant successions problem |
|------------------|----|----|----|----|----|----|----|
| 1.Open           | 0.00 | -0.22 | 0.04 | 0.15 | 0.44 | -0.82 | 0.64 |
| 2.Ulva spp.      | 0.21 | 0.00  | -0.18 | -0.05 | -0.69 | 0.71  | -0.26 |
| 3.Gigartina canaliculata | 0.30 | 0.54  | 0.00  | -0.41 | -0.10 | 1.23  | -0.54 |
| 4.Gigartina leptorynchos | 0.38 | -0.22 | 0.90  | 0.00  | -1.61 | 0.62  | 0.53 |
| 5.Gelidium coulteri | -0.29 | -0.22 | -0.37 | 0.40  | 0.00  | -1.18 | -0.10 |
| 6.Rhodoglossum affine | 2.07 | 2.05  | 2.33 | 2.57  | 2.77  | 0.00 | 1.58 |
| 7.Chthamalus fissus | 0.71 | 0.94  | 0.65 | 0.37  | 0.05 | 0.51 | 0.00 |

To obtain Nash equilibria with TEDA, one can either input the payoff matrix directly as in Figure 8.6, or upload an excel file storing the payoff matrix.
After providing the payoff matrix, one can select an algorithm to search Nash equilibria from complementarity method, Snow-Shapley, Lemke-Howson, and dual method as shown in Figure 8.7.

A parameter of distance $r$ to conduct sampling is needed to be provided if the complementarity method is adapted for Nash equilibrium searching (Figure 8.8).
Another feature of TEDA is to find the sparsest or densest Nash equilibria. As we have known, too much computation may be involved for finding the complete set of Nash equilibria. In Figure 8.9, the sparsity option provides: All (complete Nash set), Sparsest, and Densest. This option only works when choosing from complementarity method, Snow-Shapley, and dual method. The comparison of different methods for direct games in TEDA will be discussed in details.

After inputting the payoff matrix, selecting an algorithm, providing parameters if nec-
necessary, and choosing the sparsity option, TEDA will return the desired Nash equilibria and store them in the file ‘equilibrium.xls’ as following Figures 8.10, 8.11, and 8.12. The results obtained by complementarity method will be saved on sheet 1 in ‘equilibrium.xls’. In each column, the 2nd to $(n + 1)$-th elements represent a distinct Nash equilibrium, and the $(n + 2)$-th element is the number of appearance times during the searching process.

![Excel Sheet Screenshot](image.png)

Figure 8.10 Nash equilibria found by complementarity method

The results returned by the specialized Snow-Shapley method will be saved on sheet 2 in ‘equilibrium.xls’. In each column, the 2nd to $(n + 1)$-th elements represent a distinct Nash equilibrium. And notice that the specialized Snow-Shapley method gives the complete set of Nash equilibria, the result in Figure 8.11 shows that this example has one single Nash equilibrium and it is consistent with the Nash equilibrium found by complementarity method.
Figure 8.11 Nash equilibria found by Snow-Shapley method

The results returned by the specialized Lemke-Howson method will be saved on sheet 3 in ‘equilibrium.xls’. In each column, the 2nd to \((n+1)\)-th elements represent a Nash equilibrium found by starting from a distinct label point. It is often that the Nash equilibria in different columns may be repeated. The result in Figure 8.12 shows that the specialized Lemke-Howson method obtains the same Nash equilibrium from all labels and it is also consistent with the results from the previous methods.
Based on the results we have seen above, finding one Nash equilibrium by the complementarity method is computationally equivalent to solving a quadratic programming problem. The completeness of the Nash equilibria set obtained by the complementarity method can be controlled by parameters. The specialized Snow-Shapley method can always return the complete set of Nash equilibria, but the computation cost is too much for high dimension system. While the specialized Lemke-Howson can quickly find $n$ optimal strategies (may be repeated), it is quite difficult to acquire the complete solution set by using this method.

### 8.4 Stability analysis

For some found equilibria, we are interested in the stability around equilibria. To check this, we look at the eigenvalues of the Jacobian matrix of the replicator equation system at a given equilibrium. If all the eigenvalues have negative real part, the equilibrium is stable; if some of the eigenvalue have positive real part, it is unstable. Unfortunately, we could not get the stability of an equilibrium from this method if the real parts of the eigenvalues consist of negative values and zeroes.
Besides the complementarity method, Snow-Shapley, Lemke-Howson, and dual method, the connection between evolutionary game and replicator equation system provides another path to the equilibria. The ODE system can simulate the dynamics of strategies (population) if a starting state is provided. In Figure 8.14, we have the dynamics of plant succession example generated from an initial state where all species share the same coverage, \( \frac{1}{7} \).
Figure 8.14 Dynamics simulation based on replicator equations system
CHAPTER 9. SUMMARY

In this thesis, I started with looking at the optimality conditions for evolutionary games. Consider an evolutionary game defined by a symmetric fitness matrix $A \in \mathbb{R}^{n \times n}$. Let $S_n = \{x \in \mathbb{R}^n : x \geq 0, \sum_i x_i = 1\}$ be the set of all mixed strategies. Solving a direct evolutionary game is to find an optimal strategy $x^* \in S_n$ called a Nash equilibrium strategy such that

$$x^*^T A x^* \geq x^T A x^*, \quad \text{for all} \quad x \in S_n. \quad (9.1)$$

This problem is a symmetric evolutionary game. It has important applications in population genetics, where it can be used to model and study distributions of genes in given populations when they are under certain selection pressures. We explored the necessary and sufficient conditions for the equilibrium states in detail. These conditions were applied to solving direct evolutionary games.

In order to obtain the fitness matrix for solving direct games, I investigated the inverse games. An inverse game targets on recovering the payoff matrix $A$ for the evolutionary game model based on the data and replicator equations, whose parameters are the components of the payoff matrix. To obtain the estimation and inference on $A$, we first unified the different types of data sets. Then we used non-parametric spline methods to estimate the derivatives. With the smoothed data, we applied the least squares method to obtain the estimation of the payoff matrices, and we used the parametric bootstrap sampling method to obtain the inferences of the payoff matrices.

I also discussed computational schemes for solving direct games, including a specialized Snow-Shapley algorithm, a specialized Lemke-Howson algorithm, and a searching algorithm based on the solution of a complementarity problem on a simplex. The Snow-Shapley procedure is a classical algorithm for finding all extreme optimal strategies via exhausting all sub-
systems and is a purely combinatorial algorithm. The Lemke-Howson algorithm is a classical simplex type algorithm developed to search for Nash equilibria of two-player, finite-strategy games. For the evolutionary game, we make some assumptions to obtain the specialized Lemke-Howson method. Using our theoretical results, the necessary and sufficient conditions of Nash equilibrium provide complementarity conditions that lead to the complementarity method. It solves the direct evolutionary game by solving a reformulated quadratic programming problem.

After addressing the direct and inverse game, I introduced the evolutionary game dynamics. The dynamics of the replicator equation system were investigated in detail. The relationships among Nash equilibria, KKT points, and ESS were discussed. It provided another way for approaching to the Nash equilibrium by utilizing the dynamics of the replicator equation system.

Another topic I focused on is the sparsest and densest Nash equilibria in direct evolutionary games. Based on the necessary and sufficient conditions of Nash equilibrium, we derived a new algorithm, called dual method. It has the same completeness as the specialized Snow-Shapley method on Nash equilibria searching, while it does better on the computational performance.

Two numerical examples of direct evolutionary games, including a gene mutation for malaria resistance and a plant succession problem, were presented. The results about solving the direct games were applied to these examples. A solver for evolutionary games, ‘the toolbox for evolutionary dynamics analysis (TEDA)’, was shown with functions and modules, applications in inverse and direct game, stability analysis for equilibria, and dynamics analysis.
BIBLIOGRAPHY


