An Algebraic Approach to Nash Equilibria for Finite Normal Form Games

by

Ratnik Gandhi

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

Doctor of Philosophy

in

Information and Communication Technology

to

Dhirubhai Ambani Institute of Information and Communication Technology

April, 2011

This work was carried out under the supervision of
Prof. Samaresh Chatterji.

Contact
ratnik_gandhi@daiict.ac.in

Declaration

This is to certify that

1. the thesis comprises my original work towards the degree of Doctor of Philosophy in Information and Communication Technology at DA-IICT and has not been submitted elsewhere for a degree,
2. due acknowledgement has been made in the text to all other material used.

Ratnik Gandhi

Certificate

This is to certify that the thesis work titled *An Algebraic Approach to Nash equilibria for Finite Normal Form Games* has been carried out by *Ratnik Gandhi* (200521006) for the degree of Doctor of Philosophy in Information and Communication Technology at this Institute under my supervision.

Prof. Samaresh Chatterji

Acknowledgement

Doctoral studies is a journey to be an independent explorer. The journey from tame to untamed requires careful guidance and support. I was fortunate to find the needed guide in Prof. Samaresh Chatterji. I am indebted to him for being a great friend, a mentor and an advisor. My special thanks go to Prof. V P Sinha for being a source of inspiration and enlightenment. Every visit to him gives me a new point of view on concepts discussed. His course Models of Information, Communication and Computation was a life-changing experience for me and one of the reasons for my presence in the field of theory.

I wish to thank Prof. Gautam Datta for being a patient listener, critic and a friend. Discussions with him have helped me clarify finer details in my work. I am grateful to Dr. Akash Nanavati at Google Inc., for introducing me to Algorithmic Game Theory, to Prof. David A Cox at the Amherst College, Massachusetts, for guiding me through my journey in algebraic geometry and Galois Theory and also to Prof. John Dixon at the Carleton University, for helping me understand the group action. My thanks equally go to Prof. Srikrishnan Divakaran at DAIICT, and Prof. Somdeb Lahiri at IPMG, Gandhinagar, for useful and important discussions.

I am indebted to other faculty members and staff at DAIICT for the wonderful academic atmosphere explored together throughout these years. The resource that I have always enjoyed at DAIICT is its rapidly growing and multi faceted Resource Center. Many thanks to Dr. T S Kumbar for providing such a rich library experience. My work is greatly supported by LaTeX, Mathematica, Maple and Gambit softwares.

During this journey I had the privilege to be in the wonderful company of Dr. Aditya Tatu, Pratik Shah, Vikram Sorathia, Guneshwar Anand, Sunil Jardosh, Purushothaman A, Jignesh Bhavsar, Ghanshyam Parmar, Bhavesh Dharmani and other PhD colleagues. Time spent with them, on and off campus, was priceless.

My deepest appreciations and thanks goes to my wife, Ratna, for being my constant source of energy and support, and to my daughter Kahini. Thank you both for the patience and love that you have given me during the long period of my studies. I would not have been able to accomplish this work without the strong support from all my family members, friends and loved ones.

A journey is a construction to the existence. I would like to dedicate the work of this journey to my mother Jyoti and father Mahendra, my reason for being.

Table of Contents

Abstract

In this work we consider methods for computing Nash equilibria of finite normal form games that emphasize use of polynomial algebra. Nash equilibria of a game can be characterized as solutions to a system of polynomial equations that we call the *game system* ($\mathcal{GS}$). We adopt this characterization of Nash equilibria and apply polynomial algebra as a computational framework. Our work is concerned with finding all Nash equilibria given a single equilibrium (a sample equilibrium), without repeating the solution procedure for the sample equilibrium.

In the present work we consider two subclasses of finite normal form games. The class of rational payoff irrational equilibria(RPIE) games consists of the games where all the game payoff values are rational numbers while all equilibria are irrational number tuples. The class of integer payoff irrational equilibria(IPIE) games is defined similarly. The main emphasis in our work is algorithmic. We develop in detail two major algorithms required for each of the classes under consideration: a membership algorithm and an equilibria computation algorithm. We develop in detail the underlying computational techniques from polynomial algebra, and present proofs of their correctness. We compare these techniques with other algorithms and discuss their computational complexity. We also discuss approaches for constructing examples of these classes of games.

Our overall philosophy is to exploit the following: Galois groups of univariate polynomials in the ideal $\mathcal{I}$ of $\mathcal{GS}$ and a single sample solution of the $\mathcal{GS}$. We use group action by Galois groups on a sample solution to extend our knowledge about the remaining solutions of the $\mathcal{GS}$, which include all the Nash equilibria. The primary setting of our work remains Galois theory over the field of rational numbers. As

we progress to IPIE games, we use the more generalized Galois theory over commutative rings. Accordingly, several subsidiary results of an essentially algebraic nature are derived in the course of our development. We also briefly consider the possibility of games over finite fields.

For the problem of computing all the Nash equilibria of the classes of games, we present two separate but similar algorithms for RPIE and IPIE games. The algorithms work in two phases: computation of a sample solution of the $\mathcal{GS}$, followed by computation of Nash equilibria using the Galois group action. For RPIE games, in the first phase, computation of a sample solution is carried out by identifying a Gröbner basis for the ideal $\mathcal{I}$ of the $\mathcal{GS}$, while the same computation for another algorithm for IPIE games is performed with multivariate Newton Raphson method(MVNRM). The next phase – that of computing all other solutions – involves application of the Galois group over a sample solution. However, not all of these solutions correspond to Nash equilibria. Hence we resort to the Nash equilibrium verification algorithm to reject superfluous solutions and retain only the solutions corresponding to Nash equilibria.

We derive an important condition on the polynomial ideal $\mathcal{I}$ of $\mathcal{GS}$ to reduce repeated factorizations and substitutions, further reducing computational complexity of the presented algorithms. Further a condition is derived for computing equilibria of subclasses of games in closed form.

We present algorithms that use knowledge of a sample solution to compute other equilibria of the games. The presented methods do not require repeated factorizations and provide exact solutions. The work enables us to use algebraic properties and structure available in the $\mathcal{GS}$ of a game. It highlights some important interrelations of the equilibria of a game. The research reported in this work opens up interesting connections between algebraic geometry and game theory, thereby expanding the horizon of the problem of computing equilibria in game theory.

List of Symbols

$\mathbb{Z}$	Ring of integers
$\mathbb{Q}$	Field of rational numbers
$\mathbb{R}$	Field of real numbers
$\mathbb{C}$	Field of complex numbers
$\mathbb{F}$	Underlying field or ring. Specifically $\mathbb{Q}$ or $\mathbb{Z}$
$\mathbb{K}$	Field or ring extension of $\mathbb{F}$
$\mathcal{T}$	The class of RPIE or IPIE games
T	An RPIE or IPIE game. $T \in \mathcal{T}$
N	Set of players.
n	Number of players. i.e., $n = \|N\|$
S_i	Strategy set for player $i \in \{1, \ldots, n\}$
k_i	Number of strategies of player i. i.e., $k_i = \|S_i\|$
$\mathcal{K}^+$	$\sum_{i=1}^{n} k_i$
$\mathcal{K}^*$	$\Pi_{i=1}^{n} k_i$
$x^i_{j_i}$	Mixed strategy of player i for strategy $j_i \in S_i$
$A^i_{j_1,\ldots,j_n}$	Payoff received by player i when players consider their respective strategie $j_1, \ldots, j_n$
G	Galois group $Gal(\mathbb{K}/\mathbb{F})$
$\mathcal{GS}$	Game system
$\mathcal{I}$	Ideal generated by a $\mathcal{GS}$
$\sqrt{\mathcal{I}}$	Radical of an ideal $\mathcal{I}$
X	Set of equilibrium solutions of a $\mathcal{GS}$
β	Sample solution of a $\mathcal{GS}$
$\mathcal{V}$	Variety of the polynomial system
$\mathcal{R}_{\mathcal{S}}$	Interrelations of $A^i_{j_1,\ldots,j_n}$'s

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Rational Interactions and Decision Making

The study of rational competitive decision making in markets started concurrently with the formalization and study of material goods.[1] Game theory treats competitive rational decision makers as players in a game. John von Neumann and Morgenstern in their seminal work [66] introduced the game theoretic study in mathematical domain. They further defined and classified various games and analyzed different solution concepts for the games in specific classes.

An important assumption in game theory is that of perfect rationality: all the players are rational and act to maximize their respective payoffs. Though unrealistic, the assumption of is useful because it does approximate, in long term, behaviour of an individual. The assumption is also important as it provides separation between the game(system) and its players(operators) – providing a better handle to study them.

Starting off as a study of rational decision making in economics, game theory has become pervasive in other areas also. In the contemporary scenario, it is widely used for characterizing interactions between selfish agents participating in distributed systems such as the internet. Other application areas include studies of

[1] The historical account and origins of the study is noted in Myerson [67].

interactions of species in a biological system, networks, military and government policy and decision making, security and auctions. We discuss some applications here. See [71] for other applications.

Consider a peer-to-peer network, governed by multiple and distributed administrators, which lacks a central governing entity. For routing data through this network, cooperation among all the participating administrators is required [1]. A related issue is that of pricing in mobile ad-hoc networks. In the study of cooperation for forwarding data in these networks, a framework that uses game theoretic reputation or reward based payments to the participating agents is presented in [43].

Next consider the problem of replicating data of internet servers. Each server can be considered as a player which tries to minimize the web access delay and amount of data replication. The problem is modeled as a non-cooperative game in [46].

Game theory finds its applications in the military and governmental decision making. A famous example is that of the mutually agreeable destruction(MAD) strategy that increased deployment of arms during the cold war.[2] Radio network bandwidth allocation using auction [86] is another application of game theory in other domains. For details of the application of game theory in facility location, cryptography and evolutionary game theory, see [71].

In this work we consider generic finite normal form games.[3] It is important to note that any finite extensive form game can be converted to an equivalent finite normal form game. This means that the following discussion and the results in this work can be extended to the subclass of extensive form games with suitable modifications.

[2]Fore further details see The Trap, BBC Documentary Series by Adam Curtis.
[3]cf. Section 2.1.

1.1.1 Solution Concepts

From games next we move on to their solutions. In a general sense, a solution is a situation in a game in which players are satisfied with their outcomes. There are several types of solution concepts: dynamic equilibrium, competitive equilibrium, correlated equilibrium, market equilibrium and Nash equilibrium. In this work we focus on the Nash equilibrium. A Nash equilibrium is a strategy tuple in which no player has a unilateral incentive to change to any other strategy. Logically intuitive, the Nash equilibrium has become a standard solution concept for analyzing non-cooperative games. Most importantly, the existence of a Nash equilibrium is guaranteed for every finite game [68]. However, this is a pure existence result, and the proof does not indicate a method for constructing a Nash Equilibrium. Consequently, computation of Nash equilibria of finite normal form games is a natural algorithmic question.

The problem of computing Nash equilibria has gained importance in recent times. Assuming a background of the basics of game theory, we present a brief survey of various methods and computational complexity of the problem of computing Nash equilibria. Formal definitions and statements of assumed results are postponed to the next chapter.

For two player games, finding a Nash equilibrium is shown to be exponential in the number of strategies [57, 48, 60]. Gilboa and Zemel [33] present algorithmic results related to the problem of computing Nash equilibria and correlated equilibria.[4] They consider the problem of finding uniqueness of Nash equilibria and the problem of computing Nash equilibria that maximize payoff in finite normal form games and show that the problems are NP-hard. They further consider two player games and show that the same problems are NP-complete and CoNP-complete respectively.

Kaplan and Dickhaut [22] present a program for computing Nash equilibria. Fo-

[4] A correlated equilibrium is an intersection point of the payoff hyper-planes. It is a solution concept more general then the Nash equilibrium. The concept of correlated equilibrium was introduced by Robert Aumann in 1974.

takis et al.[28] consider the problem of computing best and worst pure strategy Nash equilibria support and show that the problem is NP-hard. Gairing et al.[29] consider the combinatorial structure of Nash equilibria that minimize or maximize payoff values and show that it is NP-hard to decided whether there is a payoff allocation that can be transformed to a pure Nash equilibria. Conitzer and Sandholm [13] show that it is NP-hard to determine whether Nash equilibria with certain natural properties exist in a stochastic game. They further consider the problem of counting the number of Nash equilibria in stochastic games and show that the problem is #P-hard. On the other hand Brandt et al. [6] show that the problem of computing pure strategy Nash equilibria or computing its support is NP-hard. Similar results for computing pure Nash equilibria in Graphical form games have been given in [36].

Studies of the problems of routing, installation of resources and formation of sub networks, in a network, using game theory are presented in [78, 84, 26]. One of the first such applications is that of a network modeled as a non-cooperative game. Results on such networks are reported in [49]. Fabrikant et al.[27] consider the problem of computing pure strategy Nash equilibria in a congestion game and show that the problem is PLS-complete.[5] Papadimitriou and Roughgarden [73] consider the problem of computing Nash equilibria in a multi-player congestion game. They show that Nash equilibria can be computed in polynomial time in a symmetric game, while the problem is PLS-complete for general cases.

Govindan and Wilson [37] present an algorithm for computing Nash equilibria that uses homotopy continuation method. They also discuss the computational complexity of their algorithm. Gottlob et. al.[36] discuss the algorithmic complexity of graphical form games and showed that even in very restrictive settings, determining whether a game has a pure Nash Equilibrium is NP-hard. A survey of Nash equilibria computation for bimatrix games can be found in [85]. For recent surveys on problem of computing Nash equilibria, one can refer to [77, 21, 6]. A comprehensive survey of various methods of computing Nash equilibria and their computational complexity results is available in [62].

[5]PLS is Polynomial Local Search.

Recent results show that the problem of computing a Nash equilibrium is PPAD-complete [11, 17].[6] The result makes use of Brouwer's fixed point theorem and Sperner's lemma. It suggests that the problem is likely to be hard in general, and it has given a major thrust to approximation methods. If we regard Nash equilibrium as a state of the game in which none of the players have an incentive to deviate, then in an approximate equilibrium the players have a low incentive ($\epsilon > 0$) to deviate. Algorithms for computing approximate Nash equilibria are discussed in [60, 18], while similar results in restricted settings are presented in [20, 24, 3]. Limitations of such methods are discussed in [19]. These results lead to investigations that address general methods for computing approximate Nash equilibria that run in polynomial time(called PTAS or Polynomial Time Approximation Scheme) [19, 18, 20]. However new results suggest that there is no general polynomial time approximation scheme for computing a Nash equilibrium [18].[7]

In the light of this and related results, it remains of interest to focus on restricted classes of games and develop methods for computing their Nash equilibria. This is also of value in terms of applications of game theory in particular domains. In the present work we consider a subclass of finite normal form games.

Nash equilibria of a game can be viewed as solutions to a system of equations and inequalities defined over payoffs and strategies. This system of inequalities can be converted into a system of polynomial equations that we call the *game system* ($\mathcal{GS}$). All the indeterminate variables in these polynomials are strategy profiles. The polynomials are multilinear in the indeterminate variables.

We adopt the characterization of Nash equilibria as solutions of the $\mathcal{GS}$ and apply polynomial algebra as a computational framework.[8] Note that the conversion of inequalities to equalities causes the $\mathcal{GS}$ to have more solutions then just the Nash equilibria.

[6] Polynomial Parity Arguments on Directed graphs.

[7] An excellent survey on the PPAD and approximation results of the problem of computing a Nash equilibrium is available in [77].

[8] The characterization is similar to that of [83].

The above approach has been followed by Herings and Peeters [39] and Datta [21], who use homotopy methods to compute a Nash equilibrium as a fixed point problem. They also suggest use of a Gröbner basis as an alternative method.

These existing algorithms for computing Nash equilibria, characterized as solutions of the $\mathcal{GS}$, typically iterate the procedure for a single solution to determine all the Nash equilibria. In their investigation of these algorithms, McKelvey and McLennan [62] make a significant observation:

> There is also no literature concerning how these algorithms(algorithms for computing sample equilibrium) might effectively utilize knowledge of a sample equilibrium. In the authors' experience, an important idea in organizing the analysis of a game by hand is to find one equilibrium, then ask how other equilibria might differ from this one; there is currently no substantiation of this wisdom in theory or in computational experience.

In other words, whether a method can be found for computing all the equilibria of the input game, given a single equilibrium (referred to hereafter as a sample equilibrium), without repeating the solution procedure for the sample equilibrium.

Motivated by the question raised by McKelvey and McLennan, and the complexity of computing even a single equilibrium, we consider the problem of computing *all* Nash equilibria for the *subclass* of finite normal form games. The subclass of finite normal form games that we consider have all rational(integral) payoffs and all irrational equilibria. The classes of games are called rational payoff irrational equilibria(RPIE) and integer payoff irrational equilibria(IPIE) games. We develop algorithms for computing all their Nash equilibria using a sample solution, thus providing a constructive answer to McKelvey and McLennan's question for games in these classes. We further present an algorithm for deciding membership to these classes of games.

In the view of unavailability of an efficient method for computing all equilibria of

finite normal form games, the study of structural properties of Nash equilibria has become important. The classes of games that we defined above are particularly important as they allow us to establish some relations among all the equilibria of a game with the knowledge of their Galois groups (discussed below). An example of RPIE game is first introduced by Shapley and Nash [80] and later by Nau et. al.[70]. The problem of storage of irrational equilibria of the classes of games is studied by Lipton and Markakis [60]. The classes of games could have been defined over arbitrary fields and their extensions. But, in practice the payoff values generally come from the ring of integers or the field of rational values and hence the choice of classes of games for study. The fact that irrational numbers are dense in the set of real numbers also make it possible to construct many more examples of the games.

1.1.2 An Algebraic Approach

To answer McKelvey and McLennan's question, we consider an algebraic approach via the Galois group. As is well-known, the Galois group of a polynomial acts as a permutation group on the set of roots. If a root of the polynomial is known along with its Galois group, then with the group action other conjugate roots of the polynomials can be computed. We adopt this approach and use it to develop suitable algorithm for the problem of computing all Nash equilibria of RPIE and IPIE games. In the following example, we compute the Galois group of a polynomial and show how to use the group action to produce other root of the polynomial.

Example 1. Let $f(x) = x^2 - 2 = 0 \in \mathbb{Q}[x]$. Elementary symmetric functions for f are $a = 0 = x_1 + x_2$ and $b = 2 = x_1 x_2$, where x_1, x_2 are roots of $f(x)$. If E is the Galois extension of $\mathbb{Q}$ by f, then for all $\phi \in Gal(E/\mathbb{Q})$, $\phi(a) = a$ and $\phi(b) = b$. Moreover, $f(x)$ is irreducible over $\mathbb{Q}$, so $\exists\ \phi_0 \in Gal(E/\mathbb{Q})$ such that $\phi_0(x_2) = x_1$. And so, $\phi_0(a) = \phi_0(x_1 + x_2) = a = 0 \Rightarrow \phi_0(x_1) = -x_1$.

In other words, it is possible to compute the Galois group of the polynomial without having to compute all its roots. Once the complete Galois group of $f(x)$ is known (isomorphic to $\mathbb{Z}_2$ in this case), we compute one of the roots $x_1 = \sqrt{2}$

and apply ϕ_0 to generate $x_2 = -\sqrt{2}$. A transitive Galois group action generates a single orbit. So group action produces all the roots of $f(x)$ without having to factorize it every time.

From the example 1.1.2 it is evident that, with a known Galois group and a sample root, the remaining roots of the polynomial can be computed. The idea of our algorithm for computing all equilibria of RPIE and IPIE games follows immediately from the example . Our overall philosophy is to exploit the Galois group of univariate polynomial in $\mathcal{I}$ of $\mathcal{GS}$ along with a single sample solution to extend our knowledge about the remaining solutions of the $\mathcal{GS}$, which include all the Nash equilibria. The example also suggests that knowledge of the Galois group does not presuppose knowledge of all the roots.[9] In the remaining treatment, it is therefore usually assumed that the Galois groups of irreducible univariate polynomials in the the ideal $\mathcal{I}$ of $\mathcal{GS}$ are known.

There are various approaches for computing a sample equilibrium (solution) of the given finite normal form game [62]. These approaches include use of Gröbner basis and numerical methods such as Newton-Raphson method. In this work we use Gröbner basis and Multivariate Newton Raphson method(MVNRM) for computing a sample solution of the classes of games that we consider. Both these methods are independently developed. The methods that we present are different from other methods in following way. The methods presented in [62] are general in nature and can be used for general finite normal form games. On the other hand methods presented in Chapters 4 and 5 involves steps specific to RPIE and IPIE games and ignore rational or integer solutions of the $\mathcal{GS}$. We discuss in detail the differences of our methods with previously known methods in Chapters 4 and 5.

The primary setting of our work remains Galois theory over the field of rational numbers $\mathbb{Q}$. As we progress to IPIE games we make use of Galois theory over commutative rings [10]. Accordingly, several subsidiary results of an essentially algebraic nature are derived in the course of our development.

[9] A method for computing Galois group using Tschirnhaus Transformations is presented in [32].

To our knowledge, a method for computing Nash equilibria of the classes of games with known Galois groups has not been considered earlier. An algorithm for fast decomposition of univariate polynomials, over the field of rational numbers, with a known Galois group, has been suggested in Enge and Morain [25]. The algorithm decomposes univariate polynomials with a Galois or non-Galois field extension. Segal and Ward [79] also consider use of known Galois group in the problem of computing weight distributions for irreducible cyclic codes. Use of a Galois group for computing roots of a univariate polynomial is also mentioned in [47, 7]. All these approaches consider univariate polynomials and their Galois groups. We, in this work, not only consider single multivariate polynomials, but deal with polynomial systems. Note that the method presented in this work is general and can be used to compute solutions of system of polynomial equations that are known to have their solutions outside the underlying(defining) field or rings.

Existing methods for computing a Nash equilibrium, such as the approach based on the Gröbner basis are computationally inefficient [21]. The homotopy continuation method has added drawback of providing solutions via approximation [77, 40]. The method in [3] is highly dependent on the probability distributions chosen. Our methods reduce the computational time(compared to method based on Gröbner bases) and provide exact equilibria for a subclass of RPIE and IPIE games.[10] However, finding a more efficient method for determining a single Nash equilibrium is not an essential objective of our work.

Since our algorithms for finding all the Nash equilibrium, given a sample equilibrium, are specifically designed for RPIE or IPIE games, it becomes necessary to have preliminary algorithms for determining whether input games do indeed belong to either of these classes. These algorithms are presented in Chapter 3. Though they use the characterization of Nash equilibria as solutions of the $\mathcal{GS}$, they do not require knowledge of the Galois group. We also discuss a property which reduces the complexity of the membership algorithm, and present some methods for constructing RPIE and IPIE games.

[10] For solvable groups exact and for non-solvable based on the precision used to represent the first root. cf. Proposition 6 and Proposition 14.

During the study of RPIE games we came across the following problem: Given a number stored in computer memory with finite precision, how to distinguish whether it is a rational number or irrational number. We discuss the details of the problem in Chapter 4 and a method for dealing with it is presented in Appendix A.

In this work we present algorithms for computing all equilibria of subclasses of generic finite normal form games that use knowledge of a sample solution and Galois group. The algorithms thus replace repeated substitutions and factorization of polynomials with relatively simple and computationally efficient group actions.

1.2 Organization of the Thesis

Essential game theoretic and field theoretic preliminaries are presented in Chapter 2. Underlying game model that we consider throughout this work along with examples of Galois group over the ring of integers and infinite Galois groups are also placed along with the preliminaries. The first problem that we consider is that of deciding membership to the classes of games. The membership algorithm, the condition to improve efficiency of the membership lemma, other related results and examples showing working of the membership algorithm are presented in the Chapter 3.

Chapters 4 and 5 are essentially similar in structure. Both these chapters contain algorithms for computing all equilibria of RPIE and IPIE games respectively. Related results, computational complexity of the algorithms and examples are also presented.

We discuss in detail various approaches for constructing more examples of the class of games in Chapter 6. Related issues are also discussed. Chapter 7 concludes the work with closing notes and future directions.

Appendix A presents Buchberger's Algorithm for computing Gröbner basis, Kannan Lenstra Lovasz Algorithm, [44], for minimal polynomial construction and multivariate Newton Raphson method. Appendix B on the other hand presents

the implementation of the algorithms presented in Chapters 3, 4 and 5.

1.3 Our Contribution

Following is an exhaustive listing of our contribution. The results presented are believed to be new or for which independent proofs have been derived.

Membership Decision Problem

- We present Algorithm 3.1.1 to decide whether the given input finite normal form game – with integer or rational payoff values – is a member to the class of RPIE or IPIE games or not.
- Subsection 3.1.2 presents the worst case computational complexity of Algorithm 3.1.1.
- Proposition 1 shows the correctness of Algorithm 3.1.1.
- Working of the Algorithm 3.1.1 is discussed in Subsection 3.1.1.
- During the execution of Algorithm 3.1.1, solution tuples with rational coordinate force repeated factorization and verification of each such solution tuple for a Nash equilibrium of the input game. We formulate a property that reduces number of such factorizations and verifications. The property is presented in Conjecture 1.
- Proposition 2 gives a sufficient condition on the $\mathcal{GS}$ for Conjecture 1 to be true.
- Proposition 3(Membership Lemma) gives condition on the number of univariate polynomial computation required for deciding the membership with Conjecture 1.
- Example 8 illustrates the working of Algorithm 3.1.1 with the condition in Proposition 3.

Nash Equilibria of RPIE Games

- Algorithm 4.2.1 for computing all equilibria of a RPIE game. The algorithm makes use of Gröbner basis for sample solution computation followed by group action.
- With the help of Proposition 4 and Corollary 1 we show that the class of RPIE games is empty for bimatrix games and further Algorithm 4.2.1 can not be used for computing equilibria of bimatrix games.
- A result showing correctness of Algorithm 4.2.1 is presented in Proposition 5.
- With the available finite precision technology for representing a number in computer memory, storage of an irrational equilibrium is a difficult problem. In Proposition 6 we show that for a subclass of RPIE games, their equilibria can be computed in closed form.
- An example of a 3 players 2 strategy RPIE game to show working of the Algorithm 4.2.1 is presented in Section 4.4.
- Section 4.3 discusses worst case computational complexity of Algorithm 4.2.1.

Nash Equilibria of IPIE Games

- We construct an example of Galois extension over the ring of integers $\mathbb{Z}$ in Subsection 2.4.1.
- While constructing an example of Galois group over $\mathbb{Z}$, in Subsection 2.4.1, we make the following observations:
  - For any $n \in \mathbb{Z}$ and for any $\mathbb{Z}(\sqrt{n})$, $n\mathbb{Z}(\sqrt{n})$ is not a prime or maximal ideal.
  - Irrational ring extensions of ring of integers $\mathbb{Z}$ induces non-trivial Galois groups.

- We also construct example of an infinite Galois group with finite orbits in Subsection 2.4.2.
- Proposition 7 gives a criterion for the multivariate Newton Raphson method (MVNRM) to stop. The method is used in Algorithm 5.2.2 for computing a sample solution of the input IPIE game.
- We present Algorithm 5.2.1 for computing all equilibria of an IPIE game. The algorithm makes use of MVNRM, KLL algorithm and Galois group action.
- Proposition 8 shows that, during the division of a rational factor from the polynomial system in Algorithm 5.2.1, the desired solution set with all irrational solutions of $\mathcal{GS}$ is preserved .
- We show in Proposition 9 that Algorithm 5.2.1 can not be used with games having integer payoffs and rational equilibria.
- In the light of Proposition 9, Proposition 10 guarantees that for any IPIE game its group is non-trivial.
- Convergence condition for the MVNRM is presented in Proposition 11.
- Proposition 12 presents correctness of the Algorithm 5.2.1.
- Proposition 13 shows that algorithms and algebra of RPIE and IPIE games can not be extended to work for games over finite fields.
- Similar to Proposition 6 for RPIE games, Proposition 14 presents a condition for computing equilibria of IPIE games in closed form.
- Section 5.3 discusses worst case computational complexity of Algorithm 5.2.2 and presents a bound in Proposition 15.
- Section 5.4 discusses working of Algorithm 5.2.1.

Construction of Games We present several approaches for constructing the class of RPIE and IPIE games in Chapter 6.

- Elimination ideal approach.
- Proposition 16 gives a necessary condition on the inter-relations of game payoff values with desired set as its equilibrium set.
- Liner system of equation approach.
- Elementary symmetric polynomial approach.
- Perturbation approach.

We further compute examples to show the working of these approaches and discuss issues involved.

Working Program Systems Appendix B presents programs used for implementing Algorithms 3.1.1, 4.2.1 and 5.2.1.

Chapter 2

Preliminaries

In the previous chapter we introduced the problem of computing Nash equilibria for finite normal form games. However, we shall be concerned with two subclasses of games, which are defined in this chapter. We define the game model with which we shall be working throughout, and present some game-theoretic and algebraic preliminaries. These include some ideas related to Galois theory over rings, which is further explained by an example. Our work is primarily related to finite Galois groups, but in this chapter we do briefly discuss infinite Galois groups.

Before we begin, the notation and numbering of presented results, in this work, is as follows. All the results designated as Theorem are previously known results. Our results are given as either Propositions, Corollaries or Conjectures. All definitions except Definitions 2.1.4 and 2.1.5 are standard.

2.1 Game Theory

In Game theory, games are primarily classified by means of number of strategies of players, execution of strategies at various states of the game, level of strategic cooperation between players, and the information available to players. The number of strategies of each player in a game is either *finite* or *infinite*. The execution of strategies by the players of a game is either exactly once at the beginning of the game (*strategic* or *normal form game*), or after each action of the other play-

ers(*extensive form game*). If all the players of a game possess information about other players, such as all their strategies and payoff information, then the game is called a *complete information* game. In case of partial or no information the game is called *incomplete information* game. The level of strategic collaboration amongst players categorize games as either *cooperative* or *non-cooperative*. In this work, we restrict our attention to a subclass of finite normal form complete information non-cooperative games.

Definition 2.1.1. A strategic finite normal form game T_S is a 3-tuple $\langle N, S_i, c_i : i \in N \rangle$, where, N is a non-empty finite set of players, S_i is a non-empty finite set of strategies of player i, and for each player i its payoff is defined as a function $c_i : \times_{k \in N} S_k \to \mathbb{R}$.

Each player i's mixed strategy $\Delta(S_i)$ is a probability distribution on his set of pure strategies S_i, i.e. from S_i player i chooses strategy j with probability x_j^i, where $x_j^i \in \Delta(S_i)$. A finite normal form game with mixed strategies and expected payoff $\alpha_i : \times_{k \in N} \Delta(S_k) \to \mathbb{R}$ is called a *mixed extension* T_M of the strategic game T_S.

In the analysis of games, it is useful to define a state of the game which is in some sense optimal. Such a state of the game could be regarded as an equilibrium point, i.e. if the game is played repeatedly, the state may converge to the equilibrium point. The most commonly used equilibrium point is the one introduced by John Nash in his seminal paper [69], and now known as the Nash equilibrium. In a Nash equilibrium, no player has an incentive to make a unilateral change of strategy (a more formal definition is given below). Other equilibria are also in use, such as the competitive equilibrium and the correlated equilibrium. However, the Nash equilibrium is perhaps the most intuitively appealing equilibrium concept, and in our work we shall be solely concerned with Nash equilibria. It is well known that every mixed extension of a strategic game has a Nash equilibrium, defined as follows.

Definition 2.1.2. Given the mixed extension of a strategic game, a mixed Nash equilibrium is a strategy profile $\{x_j^i\} \in \Delta(S_i)$ such that each player's mixed strategy maximizes his payoff if the strategies of the other players are held fixed.

In words, given best actions of other players, a Nash equilibrium is a strategy-tuple from which player i would not like to deviate to any of his other strategies without decreasing his payoff.

Theorem 2.1.3. *Every finite game has a mixed Nash equilibrium.*

Using analogy between compact, convex subset of $\mathbb{R}^n$ and mixed strategy profile of players, and with continuity of cost function, Nash showed that existence of a fixed point (Brouwer [23]) is equivalent to existence of an equilibrium point in the payoff function. His theorem applies specifically to any finite normal form game; a pure strategy finite normal form game need not have a pure strategy Nash equilibrium.

In our work, we will confine our attention to the following subclasses of games.

Definition 2.1.4. A finite normal form game with all its payoffs rational numbers and all the coordinates of each equilibrium tuple irrational numbers is called a rational payoff irrational equilibria (RPIE) game.

Similarly,

Definition 2.1.5. A finite normal form game with all its payoffs integer numbers and all the coordinates of each equilibrium tuple irrational numbers is called a integer payoff irrational equilibria (IPIE) game.

We will use $\mathcal{T}$ to denote whichever of these classes of games is under consideration in a specific context. We denote a game in $\mathcal{T}$ by T. It is clear that the class of IPIE games forms a subclass of RPIE games. It is significant to mention that we have to use generalizations of the standard algebra of Galois groups over fields to IPIE games. [1] [2]

[1] cf. Chapter 5 on IPIE games.

[2] For more game theoretic concepts see Osborne and Rubinstein[72].

2.2 Underlying Model

Let T be a finite normal form game with $n = |N|$ players.[3] Each player i has $k_i \geq 2$ strategies, $|S_i| = k_i$, $\mathcal{K}^* = \Pi_{i=1}^{n} k_i$ and $\mathcal{K}^+ = \sum_{i=1}^{n} k_i$. $A^i_{j_1 j_2 \dots j_n}$ denotes the payoff received by player i when each player adopts strategy j_m for $1 \leq j_m \leq k_m$ and $m = 1, \dots, n$. The probability that player i chooses strategy $j_i \in \{1, 2, \dots, k_i\}$ is denoted by $x^i_{j_i}$,

$$0 \leq x^i_{j_i} \leq 1. \tag{2.1}$$

Moreover, for each player i,

$$\sum_{j_i=1}^{k_i} x^i_{j_i} = 1. \tag{2.2}$$

Expected payoff for player i,

$$\alpha_i \;=\; \sum_{j_1=1}^{k_1} \sum_{j_2=1}^{k_2} \dots \sum_{j_n=1}^{k_n} A^i_{j_1 j_2 \dots j_n} x^1_{j_1} x^2_{j_2} \dots x^n_{j_n} \tag{2.3}$$

In a Nash equilibrium, the following holds:

$$\alpha_i \geq \sum_{j_1=1}^{k_1} \sum_{j_2=1}^{k_2} \dots \sum_{j_{i-1}=1}^{k_{i-1}} \sum_{j_{i+1}=1}^{k_{i+1}} \dots \sum_{j_n=1}^{k_n} A^i_{j_1 j_2 \dots j_{i-1} j_i j_{i+1} \dots j_n} x^1_{j_1} x^2_{j_2} \dots x^{i-1}_{j_{i-1}} x^{i+1}_{j_{i+1}} \dots x^n_{j_n},$$
$$\text{for every } j_i \in S_i \text{ and for every } i \in \{1, \dots, n\}. \tag{2.4}$$

Multiplying on both sides of (2.2) by (2.3) and equating left hand side of (2.2) to right hand side of (2.3) we get a polynomial with addition of non-negative terms (given by condition (2.4) and (2.1)) evaluating to zero. This gives us polynomial

[3] T need not be in $\mathcal{T}$. The game model discussed up to (2.5) is general and can be applied to any finite normal form game. For the remainder of the discussion $T \in \mathcal{T}$.

equations in the following form:

$$x^i_{j_i}(\alpha_i - \sum_{j_1=1}^{k_1}\sum_{j_2=1}^{k_2}\dots\sum_{j_{i-1}=1}^{k_{i-1}}\sum_{j_{i+1}=1}^{k_{i+1}}\dots\sum_{j_n=1}^{k_n} A^i_{j_1 j_2 \dots j_{i-1} j_i j_{i+1} \dots j_n} x^1_{j_1} x^2_{j_2} \dots x^{i-1}_{j_{i-1}} x^{i+1}_{j_{i+1}} \dots x^n_{j_n}) = 0,$$
$$\text{for every } j_i \in S_i \text{ and for every } i \in \{1, \dots, n\}. \tag{2.5}$$

If now $T \in \mathcal{T}$, its equilibria are constrained to be irrational. Consequently,

$$0 < x^i_{j_i} < 1. \tag{2.6}$$

Applying (2.6) to (2.5), we obtain following equations:

$$\alpha_i - \sum_{j_1=1}^{k_1}\sum_{j_2=1}^{k_2}\dots\sum_{j_{i-1}=1}^{k_{i-1}}\sum_{j_{i+1}=1}^{k_{i+1}}\dots\sum_{j_n=1}^{k_n} A^i_{j_1 j_2 \dots j_{i-1} j_i j_{i+1} \dots j_n} x^1_{j_1} x^2_{j_2} \dots x^{i-1}_{j_{i-1}} x^{i+1}_{j_{i+1}} \dots x^n_{j_n} = 0,$$
$$\text{for every } j_i \in S_i \text{ and for every } i \in \{1, \dots, n\}. \tag{2.7}$$

Depending on the context, we will refer to the system of polynomial equations in (2.5) or (2.7) as the *game system* $\mathcal{GS}$. While considering problem of deciding membership of the classes of games, the type of the input game is not known. For this reason we consider a game system of the form (2.5). In the subsequent presentation of algorithms for computing equilibria, we assume the input game $T \in \mathcal{T}$ and hence use (2.7).

Note that all Nash equilibria of a game correspond to solutions of its game system $\mathcal{GS}$, but the converse is not necessarily true. There are more solutions to the $\mathcal{GS}$ than just the equilibria. We call them non-equilibrium solutions. Throughout this work, we study Nash equilibria via the system $\mathcal{GS}$.

We write *roots* for zeros of a univariate polynomial and *solutions* for zeros of a multivariate polynomial system. In our terminology, a solution will always be a $\mathcal{K}^+$- tuple. We write *irrational Nash equilibria* for Nash equilibria with all its co-

ordinate totally mixed real-irrational numbers. The first irrational solution tuple of $\mathcal{GS}$ is called a *sample solution*. If the sample solution is an equilibrium of the input game T, it is called a *sample equilibrium*.

Note that the system of polynomial equations $\mathcal{GS}$ over complex number field has finitely many solutions [39, 21, 38].[4]

If all the equilibria, defined by system of equalities and inequalities, are denoted by E then, following notations from [63], equilibria defined only by inequalities are denoted by E'. From (2.6), it is clear that, we focus on set $X \subseteq E'$ of totally-mixed Nash equilibria.

2.3 Field Arithmetic and Galois Groups

Given a $\mathcal{GS}$ in the form (2.5) or (2.7), we now setup the polynomial algebra and Galois theory required for our approach.

We follow the standard terminology for the most part. Essential definitions are given below. Some other definitions are presented in Appendix A. In general we follow the approach given in Cox [14].

Definition 2.3.1. An extension field $\mathbb{K}$ of a field $\mathbb{F}$ is a field that contains $\mathbb{F}$ as a subfield.

Definition 2.3.2. An extension ring $\mathbb{F} \subseteq \mathbb{K}$ is called finite if the dimension of $\mathbb{K}$ as a module over $\mathbb{F}$ is finite.

For a polynomial p defined over the field $\mathbb{F}$, its splitting field $\mathbb{K}$ is the field extended by all the roots of the polynomial. All the automorphism defined over the extension

[4] The ideal $\mathcal{I}$ of the $\mathcal{GS}$ is zero-dimensional because the monomials of $\mathcal{I}$ – which are non-member to the ideal generated by leading term monomials of ideal generator polynomials – are finitely many. These monomials are called *standard monomials* of the ideal $\mathcal{I}$ with respect to some lexicographical order. In other words, finite solutions of ideal is possible if and only if for each indeterminate x_i of the polynomial ring there is a term t in leading monomial of Gröbner basis such that $t = c \cdot x_i^n$. i.e., at least one member of Gröbner basis has its leading term as pure power of x_i.

field $\mathbb{K}$ that fix each element of base field $\mathbb{F}$, forms a group called Galois group. Action of a Galois group on roots of a polynomial results in a permutation of the roots.

Definition 2.3.3. Let G be a group and X be a set. Then an action of G on X is a function of the form $G \times X \to X$.

Due to Galois correspondence, we have one-one relation between sub-group of Galois group and sub-field extensions of splitting field of a polynomial. We shall be specifically interested in the following situation.

Definition 2.3.4. Let $\mathbb{K} \supset \mathbb{F}$ be a finite extension of a field $\mathbb{F}$. Then the Galois group $G = Gal(\mathbb{K}/\mathbb{F})$ is the set

$$G = \{\sigma : \mathbb{K} \to \mathbb{K} \mid \sigma \text{ is an automorphism, } \sigma(a) = a \text{ for all } a \in \mathbb{F}\}.$$

Definition 2.3.4 will be considered while we discuss the class of RPIE games. Galois theory has recently been generalized to work with polynomials defined over rings. For discussing the class of IPIE games we need the generalization. Definition 2.3.4 can be generalized as follows.

Definition 2.3.5. [42] Let $\mathbb{K}$ be a finite extension of commutative ring $\mathbb{F}$, i.e. $\mathbb{F}$ is subring of $\mathbb{K}$. Let G be a finite group acting as $\mathbb{F}$-algebra(ring) automorphisms on $\mathbb{K}$. Then we define $\mathbb{K}^G$ as the subring

$$\mathbb{K}^G = \{s \in \mathbb{K} \mid \forall \sigma \in G, \sigma s = s\},$$

and say that $\mathbb{K}$ is a Galois extension of $\mathbb{F}$ with group G, if

- $\mathbb{K}^G = \mathbb{F}$, and
- for any maximal ideal $\mathfrak{m}$ in $\mathbb{K}$ and any $\sigma \in G \backslash \{1\}$, there is an $s \in \mathbb{K}$ such that $\sigma s - s \notin \mathfrak{m}$.

An example for constructing Galois group of an irrational extension of the ring of integers is presented in Section 2.4. For detailed discussion of concepts related to polynomial algebra refer Appendix A.

When a Galois group acts on subset of roots, due to the group action we get the other elements of the set of roots.[5] For generating all the elements of a set of roots, transitivity of the Galois group is necessary.

Definition 2.3.6. A subgroup $H \subset S_n$ is transitive if for every pair of elements $i, j \in \{1, 2, \ldots, n\}$, there is $\tau \in H$ such that $\tau(i) = j$.

Below result allows us to see irreducibility of a polynomial as transitivity of its Galois group.

Theorem 2.3.7. *Let $\mathbb{K}$ be the splitting field of a separable polynomial $f \in \mathbb{F}[x]$ of degree n. Then the subgroup of S_n corresponding to $Gal(\mathbb{K}/\mathbb{F})$ is transitive if and only if f is irreducible over $\mathbb{F}$. i.e. f is irreducible if and only if $Gal(\mathbb{K}/\mathbb{F})$ acts transitively on the roots of f.*

For proof of this theorem see [14].

Action by all the elements of a group on an element of a set generates a set called orbit of that element. Formally,

Definition 2.3.8. For every $x \in X$ we put $Gx = \{gx : \forall g \in G\}$, and call it the orbit of x under action of G, or simply G-orbit of x.

Orbit of an element under Galois group is called Galois-orbit of the element. If G acts transitively on X then there is only one G-orbit, X itself.

2.4 Extensions and Generalizations of Galois Groups

In this section we present some ideas that are not elementary. Though not new, these ideas are independently developed.

2.4.1 Example of a Galois Group Over the Ring of Integers

Following is an example of a non-trivial ring extension that produces a non-trivial Galois group given by Definition 2.3.5.

[5] Not necessarily distinct.

Let $\mathbb{F} = \mathbb{Z}$ and $\mathbb{K} = \mathbb{Z}(\sqrt{2})$ then for all primes $p \neq 2$, $p\mathbb{Z}(\sqrt{2})$ is maximal ideal of $\mathbb{K}$, and the Galois group for this ring extension $\mathbb{K}/\mathbb{F}$ is isomorphic to $\mathbb{Z}_2$.[6] Choice of $p\mathbb{Z}(\sqrt{2})$ as maximal ideal can be justified as follows:

It is known that all prime ideals $p\mathbb{Z}$ of $\mathbb{Z}$ are its maximal ideals. For finding maximal ideal of $\mathbb{Z}(\sqrt{2})$, natural way to progress is to consider $p\mathbb{Z}(\sqrt{2})$ of $\mathbb{Z}(\sqrt{2})$ as a candidate. Other way to reach this choice is, $\mathbb{Z}(\sqrt{2})$ is isomorphic to the ring of Gaussian Integers $\mathbb{Z}(i)$.[7] And it is known that an $a \in \mathbb{K}$ is a prime element in Euclidean ring $\mathbb{K}$ if and only if the ideal $\langle a \rangle$ is maximal in $\mathbb{K}$. We choose a prime element $p(1+1\sqrt{2}) \in \mathbb{Z}(\sqrt{2})$.This prime element generates an ideal $p\mathbb{Z} + p\mathbb{Z}(\sqrt{2})$. And so the choice of prime ideal. We must verify whether $p\mathbb{Z}(\sqrt{2})$ is indeed a maximal ideal. But, first we check whether it is an ideal of $\mathbb{K}$ or not.

All the elements in $p\mathbb{Z}(\sqrt{2})$ are of the form $mp + np\sqrt{2}$, for $m, n \in \mathbb{Z}$. It is easy to see that $p\mathbb{Z}(\sqrt{2})$ is closed under addition, closed under multiplication, commutative and every element $a \in \mathbb{Z}(\sqrt{2})$ is *absorbed* inside $p\mathbb{Z}(\sqrt{2})$.

Maximality of ideal $p\mathbb{Z}(\sqrt{2})$ can be verified by two methods. First, argue that there is no proper ideal of $\mathbb{Z}(\sqrt{2})$ that contains $p\mathbb{Z}(\sqrt{2})$. Second, create quotient ring of $\mathbb{Z}(\sqrt{2})$ and $p\mathbb{Z}(\sqrt{2})$ and see whether it forms a field or not.[8] We consider the second method.

All the elements in $\mathbb{Z}(\sqrt{2})$ are of the form $\{a+b\sqrt{2}\}$, and elements of $p\mathbb{Z}(\sqrt{2})$ are of the form $\{mp+np\sqrt{2}\}$, for $a, b, m, n \in \mathbb{Z}$. The coset of the quotient $\mathbb{Z}(\sqrt{2})/p\mathbb{Z}(\sqrt{2})$ has following elements:

$$\begin{array}{llll} \{0, & 1, & \ldots, & p-1, \\ \sqrt{2}, & 1+\sqrt{2}, & \ldots, & (p-1)+\sqrt{2}, \\ \vdots & \vdots & & \vdots \\ (p-1)\sqrt{2}, & 1+(p-1)\sqrt{2}, & \ldots, & (p-1)+(p-1)\sqrt{2}\}. \end{array} \tag{2.8}$$

[6] For the reason of $p \neq 2$, cf. Proposition 1.

[7] It can be shown that $\mathbb{Z}(\sqrt{2})$ in fact is an Euclidean Ring with $d = a^2 + b^2$ for all $a + b\sqrt{2} \in \mathbb{Z}(\sqrt{2})$.

[8] Quotient ring of a commutative ring with its maximal ideal is a field.

The quotient structure above has total of p^2 elements for any prime p, and so are likely candidates to be finite fields. For example, if $p = 3$, then the elements of the quotient structure $\mathbb{Z}(\sqrt{2})/3\mathbb{Z}(\sqrt{2})$ are

$$\{0, 1, 2, \sqrt{2}, 2\sqrt{2}, 1+\sqrt{2}, 1+2\sqrt{2}, 2+\sqrt{2}, 2+2\sqrt{2}\}.$$

It can be verified that $\mathbb{Z}(\sqrt{2})/3\mathbb{Z}(\sqrt{2})$ is a finite field with 9 elements. The claim can be verified to be true for $\mathbb{Z}(\sqrt{2})/p\mathbb{Z}(\sqrt{2})$ in general.

On the other hand for $p = 2$, elements of $\mathbb{Z}(\sqrt{2})/2\mathbb{Z}(\sqrt{2})$ are $\{0, 1, \sqrt{2}, 1+\sqrt{2}\}$. This is not a field because $\sqrt{2} \cdot \sqrt{2} = 2 \; mod \; 2 \equiv 0$, in fact it is not even integral domain. This argument generalizes to:

Remark 1. *For any $n \in \mathbb{Z}$ and for any $\mathbb{Z}(\sqrt{n})$, $n\mathbb{Z}(\sqrt{n})$ is not a prime or maximal ideal.*

Proof. For any ring isomorphic to Gaussian integers, all of its maximal ideals are generated with the prime elements in the ring. But $n = \sqrt{n} \cdot \sqrt{n}$ is not a prime element in the ring $\mathbb{Z}(\sqrt{n})$. And so the result follows. □

This example shows that $p\mathbb{Z}(\sqrt{2})$ is indeed a maximal ideal of $\mathbb{Z}(\sqrt{2})$. To show that $\mathbb{Z}(\sqrt{2})/\mathbb{Z}$ is a Galois extension, let $\sigma \in G(\mathbb{K}/\mathbb{F})\backslash\{1\}$. For every maximal ideal pertaining to a different p, we can always choose an element $s \in \mathbb{K}$ which is a non-multiple or co-prime to p, such that $\sigma s - s \notin p\mathbb{Z}(\sqrt{2})$.[9] This shows that $\mathbb{K}/\mathbb{F}$ is indeed a Galois extension and $G = Gal(\mathbb{K}/\mathbb{F})$ a Galois group. Note that, above results can be generalized by replacing $\mathbb{Z}(\sqrt{2})$ with $\mathbb{Z}(\sqrt{p})$ for $\mathbb{K}$, or replacing $\sqrt[2]{}$ with $\sqrt[m]{}$, for $m \geq 3 \in \mathbb{Z}$. We call ring extensions by irrational numbers as irrational ring extensions. Following result generalizes the example discussed above.

Remark 2. *Irrational ring extensions of ring of integers $\mathbb{Z}$ induce non-trivial Galois groups.*

Proof. Follows from the discussion above. □

[9]For example, $p = 3, \mathbb{K} = \mathbb{Z}(\sqrt{2})$ and $\sigma \in Gal(\mathbb{K}/\mathbb{Z}) \cong \mathbb{Z}_2$, let $s = 5 + 7\sqrt{2}$ then for non identity σ, $\sigma(5 + 7\sqrt{2}) - (5 + 7\sqrt{2}) = -14\sqrt{2} \notin 3\mathbb{Z}(\sqrt{2})$.

2.4.2 Infinite Galois Group

In this work we are concerned with finite Galois groups. But for the completeness of discussion, we present an example of infinite Galois group that produces finite group-orbits.

Let $f \in \mathbb{Q}[x, y]$ be a bivariate polynomial of the form

$$f(x, y) = x^2 + y^2 = 0 \in \mathbb{Q}[x, y].$$

Let $\mathbb{K}$ be a splitting field extension of f over $\mathbb{Q}$. Over $\mathbb{K}$, f has following factorization:

$$f(x, y) = (x + iy)(x - iy) = 0.$$

Ignoring the trivial case of exactly one solution, a multivariate polynomial has infinitely many solutions in its separable closure. Of infinitely many solutions let one of the solution of f be: $x - iy = 0 \Rightarrow x = iy$ and for some $y = \sqrt{2} \in \mathbb{K}$, $x = i\sqrt{2}$. i.e. $(x, y) = (i\sqrt{2}, \sqrt{2})$.

The minimal polynomial of $(i\sqrt{2}, \sqrt{2})$ is $(x^2 + 2)(y^2 - 2) = 0$, and produces the field extensions $\mathbb{Q}_{1x} = \mathbb{Q}(i\sqrt{2})$ and $\mathbb{Q}_{1y} = \mathbb{Q}(\sqrt{2})$ of degree 2 for x and y respectively. $\mathbb{K}$ is made up of such infinitely many finite extensions of $\mathbb{Q}$ and so $\mathbb{K}$ is called *finitely generated extension* of $\mathbb{Q}$.

Let the Galois group corresponding to the field extension $\mathbb{Q}_{1x}/\mathbb{Q}$ be G_{1x}. Then it is clear that, G_{1x} has two elements and group action of G_{1x} over the root $i\sqrt{2}$ will produce a finite group-orbit. Similarly for G_{1y}.

This argument can be repeated for infinitely many different values of y, each time producing a finite group-orbit. We know that $\mathbb{K}$ is the field containing all the solutions of polynomial f and their conjugates, so it is easy to see that Galois group $G = \mathbb{K}/\mathbb{Q}$ will have infinitely many elements. This can be illustrated by the figure 2.1, where all the Galois groups are finite. Note that the field corresponding to the group – with all Galois groups as it subgroup – is infinite. For more

discussion on the infinite Galois groups from permutation groups point of view see [2].

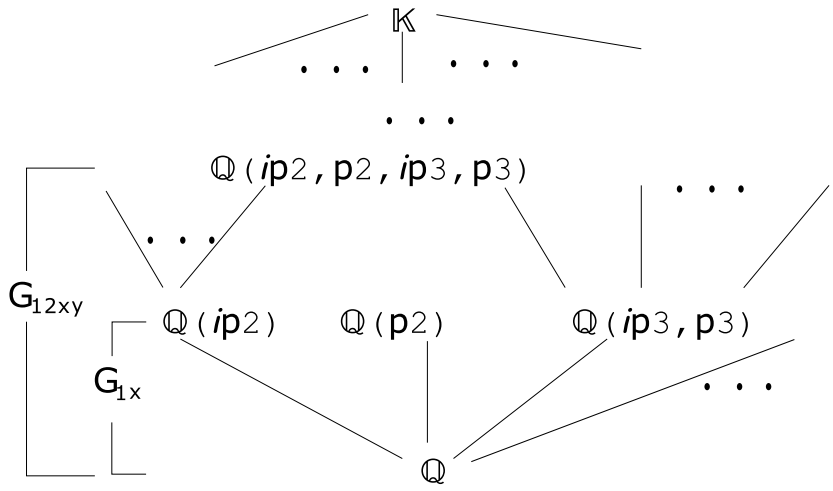

Figure 2.1: Infinite field extension and corresponding Galois group.

2.5 Discussion

With required definitions, preliminaries and game model we are ready to take upon the following question: given an input game how can we decide whether it is a member to the classes of games that we defined in the present chapter. In the next chapter we answer this question with an algorithm. Games with integer payoffs and irrational equilibria – requiring Galois groups over rings – are discussed in Chapter 5.

Chapter 3

Membership

We defined the classes of IPIE and RPIE games in the preceding chapter. The underlying model for the classes of games was also presented. In this chapter, we consider the problem of deciding membership of a finite normal form game to the classes of RPIE and IPIE games. We present an algorithm for deciding membership, and discuss related results and the computational complexity of the algorithm. We notice that in case the $\mathcal{GS}$ has certain regularity properties, the efficiency of the presented algorithm is considerably improved. This issue is discussed in detail. Finally, examples are presented to display the working of the algorithm with and without regularity property. The membership decision is necessary for subsequent chapters that present algorithms for computing equilibria of RPIE and IPIE games.

3.1 Method

The games that we consider are required to have either rational or integer payoff values and all irrational Nash equilibria. With this property, an intuitive approach to answer the problem of deciding membership is as follows: given an input game, characterize all its equilibria as solutions to the $\mathcal{GS}$ of the form (2.5); construct a univariate polynomial – for each indeterminate variable – in the Gröbner basis of the $\mathcal{GS}$; determine whether for each indeterminate variable its univariate polynomial has linear factors over the field of rational numbers or not. It turns out that

this approach is successful, though each of the steps has to be handled carefully, and requires rigorous justification.

Note that the membership of the class of IPIE games can also be decided by considering irreducibility of univariate polynomials over $\mathbb{Q}$. If a polynomial has all its factors linear over $\mathbb{Q}$, then its roots are either integer or rational. If the polynomial has no linear factors over $\mathbb{Q}$ then the roots are either irrational or complex. By [68], every game has a mixed strategy Nash equilibrium. In the case of an IPIE game, this means that there is at least one irrational root of the polynomial. The condition of checking irreducibility of each univariate polynomial over $\mathbb{Q}$ – rather than $\mathbb{Z}$ – lets us use field algebra, providing a richer set of tools. Thus, a single algorithm covers both RPIE and IPIE games; the payoff values suffice to determine whether the input is an RPIE or IPIE game.

In case a univariate polynomial in the Gröbner basis has some linear factors over $\mathbb{Q}$, we must verify whether the solutions – determined by the corresponding rational roots – are Nash equilibria of the input game or not. If any of these solutions turns out to be an equilibrium then the game is a non-member, otherwise it is. For polynomial irreducibility over the set of integers we can use Eisenstein's criterion [74]. But this is not a necessary and sufficient condition, and in any case, we need more tools to decide irreducibility over $\mathbb{Q}$. For this purpose, we make use of the univariate factorization algorithm over $\mathbb{Q}$ given in [31]: an advantage of this algorithm is that it has polynomial time complexity.

For constructing a univariate polynomial from the multivariate system $\mathcal{GS}$, we use a Gröbner basis of the $\mathcal{GS}$. The Gröbner basis construction can be considered as a preprocessing step for an input, and the triangular form generated in the process may be needed in the sequel. If the input game turns out to be a member game in either of the classes, then the Gröbner basis is in fact utilized further for computing all equilibria of the input game with Algorithms 4.2.1 and 5.2.1 presented in the following chapters.

We commence the attack on the membership decidability problem by checking the

payoff values of the input game. If the payoff values are non-integer(non-rational) then we declare the input game a non-member of the class of IPIE(RPIE) games. Otherwise we use Algorithm 3.1.1.

Algorithm 3.1.1 Algorithm for deciding membership to the classes of RPIE and IPIE games.
Input : A finite normal form game characterized as $\mathcal{GS}$ with coefficients from $\mathbb{Z}$ or $\mathbb{Q}$.
Output: Member or Non-member Decision.

1: **for** each indeterminate variable ($i = 1$ to $\mathcal{K}^+$) **do**
2: Apply the Buchberger's Algorithm with the lexicographic order ($x_i \prec x_j), \forall j \neq i$ and compute a univariate polynomial g_i in x_i from the triangular form of the Gröbner basis of $\mathcal{GS}$.
3: For the univariate polynomial produced in Step 2, check its irreducibility over $\mathbb{Q}$. {In case any g_i has all linear factors over $\mathbb{Q}$, the algorithm must stop immediately declaring the input game a non-member.}
4: **if** g_i has at least one linear factor over $\mathbb{Q}$ **then**
5: Compute root corresponding to each linear factor of g_i.
6: Substitute each root in the triangular form of the Gröbner basis and compute a complete solution tuple corresponding to the root.
7: **if** the solution tuple verifies to be a Nash equilibrium of the input game **then**
8: Declare the input game a non-member and stop.
9: **end if**
10: **end if**
11: **end for**
12: Declare the input game a member.

Change of lexicographic order to produce univariate polynomial in each indeterminate in Step 2 of Algorithm 3.1.1 can be justified by the following quote from [8]:

> The elimination property of Gröbner bases guarantees that, in case G has only finitely many solutions, G contains a univariate polynomial in x. (Note that, here, we use the lexicographic order that ranks y higher than x. If we used the lexicographic order that ranks x higher than y then, correspondingly, the Gröbner basis would contain a univariate polynomial in y.) [...] It can be shown that reduced Gröbner bases

(with finitely many solutions) contain exactly one univariate polynomial in the lowest indeterminate.

Next we present the correctness of the membership Algorithm 3.1.1.

Proposition 1. *Given an input game with integer(rational) payoff values, Algorithm 3.1.1 correctly determines whether it belongs to class IPIE(RPIE).*

Proof. Input to the Algorithm 3.1.1 is a finite normal form game characterized as $\mathcal{GS}$ of the form (2.5).[1] Consequently, all the coefficients of the $\mathcal{GS}$ must be either integral or rational. Then the loop of Steps 1-11 executes. Next step is to check whether all the equilibrium solutions of the input game are irrational or not. For checking the irrationality of each indeterminate variable in the $\mathcal{GS}$, we must check whether its univariate polynomial has all its factors irreducible over $\mathbb{Q}$ or not. To obtain a univariate polynomial in each indeterminate variable the Buchberger's algorithm is called using a different lexicographic ordering each time. Due to the finiteness of number of equilibrium solutions of the input game and its corresponding zero-dimensional polynomial ideal, we are guaranteed to get a univariate polynomial every time [8].

For Steps 3 and 4, the univariate factorization algorithm from [31] is used. Thus, whenever the univariate polynomial has a linear factor over $\mathbb{Q}$, Step 5 is reached.

Corresponding to each linear factor of a univariate polynomial a solution tuple is constructed. These solutions are further verified for Nash equilibrium of the input game. If one of these solutions is a Nash equilibrium, then the method stops with declaring the input game a non-member at Steps 7 and 8.

With the finiteness of the following: degree bound of the degrees of the $\mathcal{GS}$, the degree of each univariate polynomial and number of strategies, we are guaranteed to either stop at Step 8 or reach Step 11.

[1] We recall that the type of games that we consider in this work are known to have finitely many equilibrium solutions [39, 21, 38].

If we reach Step 11, then none of the univariate polynomials of any variable has contributed a Nash equilibrium with a rational coordinate. Hence, any Nash equilibria solution could only have non-rational coordinates. However, since by Nash's theorem, there is at least one Nash equilibrium, the game has to be either IPIE or RPIE. □

Next we present examples to show working of the Algorithm 3.1.1.

3.1.1 Examples

Example 2 (Matching Pennies)**.** Game of matching pennies can be described by Table 3.1. It is known that the game does not have a pure strategy Nash equilibrium. Also, it is know that the game has exactly one mixed strategy Nash equilibrium $(\frac{1}{2}, \frac{1}{2})$. The game is not an RPIE/IPIE game. We verify this using Algorithm 3.1.1.

		Player 2	
		H	T
Player 1	H	1, -1	-1, 1
	T	-1, 1	1, -1

Table 3.1: Payoff table of 2-player 2-strategy zero-sum Matching Pennies game. A coin is tossed twice for deciding the output. If both the toss has matching result, i.e., Head-Head or Tail-Tail, Player 1 gains one penny. Otherwise, Player 2 gains 1 penny. A penny loss for one player is a penny gain for the other player.

Following is the $\mathcal{GS}$ of the game.

$$\begin{aligned} 2(-1+x)x(-1+2y) &= 0 \\ -2(-1+x)x(-1+2y) &= 0 \\ -2(-1+2x)(-1+y)y &= 0 \\ 2(-1+2x)(-1+y)y &= 0 \end{aligned} \tag{3.1}$$

The Gröbner basis for the $\mathcal{GS}$ in (3.1) is:

$$\left\{y - 3y^2 + 2y^3, y - 2xy - y^2 + 2xy^2, -x + x^2 + y - y^2\right\} \tag{3.2}$$

Univariate polynomial $f = y - 3y^2 + 2y^3 = 0$ has roots $y = 0, 1, \frac{1}{2}$. The polynomial f has complete linear factorization over $\mathbb{Q}$ and so the algorithm 3.1.1 stops, declaring the Matching Pennies game a non-member to IPIE/RPIE class games.

For the completeness of the discussion, we ignore the fact that f has all linear factors over $\mathbb{Q}$ and show further how the Nash equilibrium verification algorithm can be employed to declare the non-membership of the Matching Pennies game.

Substitution of all these factors in (3.2) generates following solution tuples of the $\mathcal{GS}$.

$$(x, y) = \left\{(0,0), (0,1), (1,0), (1,1), (\frac{1}{2}, \frac{1}{2})\right\} \tag{3.3}$$

We verify the tuples with rational roots for Nash equilibrium with the Nash equilibrium verification algorithm:

	Probability and Payoff at Strategy 1	Probability and Payoff at Strategy 2
Player 1	0,-1	1,-1
Player 2	1,1	0,1

	Probability and Payoff at Strategy 1	Probability and Payoff at Strategy 2
Player 1	1/2,0	1/2,0
Player 2	1/2,0	1/2,0

Table 3.2: Matching Pennies strategy payoff table. Entry in each cell indicates a probability that a player assigns to his particular strategy and corresponding payoff received for the assignment.

When testing for tuple $(x, y) = (0, 1)$, from Table 3.1 it is clear that the payoff received is -1 and 1 for players 1 and 2 respectively. This entry is reported in first column of the Table 3.2. It is clear that the solution tuple $(0, 1)$ does not benefit Player 1 and so he has incentive to deviate and the tuple $(0, 1)$ is not a Nash equilibrium of the game. The same is true for $(0, 0), (1, 0)$ and $(1, 1)$. They are not Nash equilibria of the game. But verification reveals that the rational solution tuple $(1/2, 1/2)$ turns out to be the Nash equilibrium of the game. And so the game is non-member.

Example 3. Consider a 2 player finite normal form game with rational payoff values. The game is defined by the payoff table 3.3. If we represent probability that player 1 chooses strategy a and b by x_1 and x_2 respectively, then probability

	A	B
a	1, 2	0, 3
b	2, 0	0, 1
c	0, 1	4, 0

Table 3.3: Payoff table of a 2-player finite normal form game. Player 1 has 3 strategies a, b and c while player 2 has two strategies A and B. Entry in each cell of the payoff table indicates player 1 and 2's payoffs for their respective strategies.

of choosing strategy c is $1 - x_1 - x_2$. Similarly, for player 2, the probability of choosing strategy A is y_1 and B is $1 - y_1$. The corresponding $\mathcal{GS}$ is (following 2.7):

$$\begin{aligned}
x_1(4 - 5y_1 + x_1(-4 + 5y_1) + x_2(-4 + 6y_1)) &= 0 \\
x_2(2(-1 + x_2)(-2 + 3y_1) + x_1(-4 + 5y_1)) &= 0 \\
-(-1 + x_1 + x_2)(-4x_1 - 4x_2 + 5x_1y_1 + 6x_2y_1) &= 0 \\
-(-1 + 2x_1 + 2x_2)(-1 + y_1)y_1 &= 0 \\
(-1 + 2x_1 + 2x_2)(-1 + y_1)y_1 &= 0
\end{aligned} \tag{3.4}$$

Computing a Gröbner basis with the lexicographic order $y_1 \prec x_2 \prec x_1$, we get:

$$\begin{aligned}
&\{-8y_1 + 30y_1^2 - 37y_1^3 + 15y_1^4, -10y_1 - 4x_1y_1 + 25y_1^2 + 4x_1y_1^2 - 15y_1^3, \\
&-50y_1 - 8x_1y_1 + 8x_1^2y_1 + 125y_1^2 - 75y_1^3, \\
&12y_1 - 4x_2y_1 - 27y_1^2 + 4x_2y_1^2 + 15y_1^3, \\
&-32x_1 + 32x_1^2 + 32x_1x_2 - 250y_1 + 625y_1^2 - 375y_1^3, \\
&32x_1 - 32x_1^2 - 32x_2 + 32x_2^2 + 466y_1 - 1111y_1^2 + 645y_1^3\}.
\end{aligned} \tag{3.5}$$

Factoring univariate polynomial $f = 15y_1^4 - 37y_1^3 - 30y_1^2 - 8y_1 = 0$ we get its roots $y_1 = 0, 1, \frac{2}{3}, \frac{4}{5}$.

It is clear that each solution of the game $\mathcal{GS}$ contains a root of y_1 that is a rational number. The game must have at least one Nash equilibrium solution. And so the equilibrium solution tuple contains a rational root. Thus the game is a non-member.

3.1.2 Computational Complexity

Running time of the Algorithm 3.1.1 is primarily dominated by the preprocessing task of constructing univariate polynomials. The Buchberger's algorithm for constructing univariate polynomials takes doubly exponential time in the number of indeterminate variables $\mathcal{K}^+$, making worst case complexity of the algorithm $\mathcal{O}(\mathcal{K}^+ \cdot 2^{2^{\mathcal{K}^+}})$.

The algorithm to factorize a univariate polynomial over $\mathbb{Q}$ takes $\mathcal{O}(n^{10}+n^8 \log^2 A)$ time [31], where n is degree of the univariate polynomial and A is its max-norm, i.e. $A = ||a||_\infty = \max\{|a_i| : 0 \leq i \leq n\}$ where a_i's are coefficients of the monomials of the univariate polynomial f. In worst case, the factorization algorithm is also called for total of $\mathcal{K}^+$ times. The Algorithm to verify a solution for Nash equilibrium runs in polynomial time in $\mathcal{K}^+$. The $\mathcal{GS}$ is of finite size in terms of the degree of individual polynomials, the total number of indeterminate variables and norm of the polynomials. Keeping aside the running time of Buchberger's algorithm, membership decidability runs in polynomial time in $\mathcal{K}^+$.

3.2 Membership Lemma

We reanalyze the roots generation sequence in the Algorithm 3.1.1. If at any stage of factorizations, a univariate polynomial in the Gröbner basis of $\mathcal{GS}$ has a linear factor, i.e. a rational root, of which corresponding solution is a Nash equilibrium, then the game is a non-member. In other words, a solution tuple with at least one rational coordinate must be verified to be a Nash equilibrium. Consequently, if we are guaranteed that the substitution of an irrational root produces all irrational roots in subsequent substitution, then the repeated factorization and verification of the roots in Algorithm 3.1.1 can be reduced. A condition which guarantees this is called a *membership condition*.

In this section we present a candidate membership condition that improves running time of the Algorithm 3.1.1. The result primarily utilizes an algebraic property of

the ideal $\mathcal{I}$ of $\mathcal{GS}$.

3.2.1 Regularity Property of the $\mathcal{GS}$

With reference to the game model presented in Section 2.2 of Chapter 2, we propose the following:

Conjecture 1. *Let $\mathcal{GS}$ be a system of polynomial equations, with zero dimensional ideal $\mathcal{I}$, of the form*

$$\mathcal{GS} = \{f_i \in \mathbb{F}[x_1, x_2, \ldots, x_k] \mid i \in \{1, \ldots, k\}\},$$

where $\mathbb{F}$ is either $\mathbb{Z}$ or $\mathbb{Q}$. A Gröbner basis of the $\mathcal{GS}$ is

$$\mathcal{GB}_{\mathcal{GS}} = \{g_j \in \mathbb{F}[x_1, x_2, \ldots, x_k] \mid j \in \{1, \ldots, m\}\}.$$

Let $g_t \in \mathbb{F}[x_t]$, for $t \in \{1, \ldots, m\}$, be a univariate polynomial in the $\mathcal{GB}_{\mathcal{GS}}$. Suppose $g_t(\alpha_{t_p}) = 0$ for some root $\alpha_{t_p} \notin \mathbb{F}$. The new Gröbner basis is

$$\mathcal{GB}'_{\mathcal{GS}} = \{g'_j \in \mathbb{F}[x_1, \ldots, x_{t-1}, \alpha_{t_p}, x_{t+1}, \ldots, x_k] \mid j \in \{1, \ldots, m\}\}.^2$$

Let $g'_q \in \mathcal{GB}'_{\mathcal{GS}}$ such that $g'_q \in \mathbb{F}[x_q]$. If $g'_q(\alpha_{q_r}) = 0$ then $\alpha_{q_r} \notin \mathbb{F}$, i.e. if α_{q_r} is some root of g'_q then α_{q_r} extends $\mathbb{F}$.

In what follows, we argue for the plausibility of Conjecture 1 using a constructive approach. But first we see with the following example that shows the conclusion of Conjecture 1 is true for polynomials that does not arise from a game($\mathcal{GS}$).

Example 4. Consider polynomials

$$\begin{aligned} f_1 &= xy^2 + 6x - y^2 - 6 - yx^2 - y \\ f_2 &= yx^2 + 6y - x^2 - 6 - xy^2 - x \end{aligned} \tag{3.6}$$

[2]Note that substituting α_{t_p} in $\mathcal{GS}$ need not extend $\mathbb{F}$. For example consider the system $f = z^2x^2 - 19 = 0$, $g = z^2 - 19 = 0$. Substituting $\sqrt{19}$ in f does not extend $\mathbb{Q}$. If f is of the form $zx^2 - 19 = 0$, then the substitution induces $\mathbb{Q}(\sqrt{19})$.

defined over $\mathbb{Q}[x, y]$. The univariate polynomial of y in a Gröbner basis, with lexicographical order $y \prec x$ has roots $y = 2, 3, \frac{1}{2}(1 \pm i5)$. Substituting $y = \frac{1}{2}(1 - i5) \notin \mathbb{Q}$ in the triangular form of the Gröbner basis and solving for x, we get $x = \frac{1}{2}(5 \pm \sqrt{11 - 40i})$. Substituting $y = 2$, on the other hand gives solution $x = 2, 3$. This shows that polynomials f_1 and f_2 follow Conjecture 1.

Conjecture 1 is not true in general for all polynomial system. We see this with following example.

Example 5. Let $f_1, f_2 \in \mathbb{Q}[x, y]$ be

$$\begin{aligned} f_1 = x^2y^2 - 19 = 0 \\ f_2 = y^2 - 19 = 0. \end{aligned} \tag{3.7}$$

System in (3.7) has a zero-dimensional ideal. Its Gröbner basis is $\{y^2 - 19, x^2 - 1\}$. Substituting $y = \sqrt{19}$ in f_1, we get rational value of x. This shows that Conjecture 1 is not true in general.

Example 6. Now suppose we have a system

$$\begin{aligned} f_1 = x^2 - 3z^2 = 0 \\ f_2 = y - 2z = 0 \\ f_3 = z^3 - 3 = 0 \end{aligned} \tag{3.8}$$

over $\mathbb{Q}$. It is clear that f_3 is irreducible over $\mathbb{Q}$. Moreover x and y in f_1 and f_2 can respectively be written in the form of z.

$$\begin{aligned} x^2 = 3z^2 = g_1(z) \\ y = 2z = g_2(z) \end{aligned} \tag{3.9}$$

Note that degrees of g_1 and g_2 are non-zero and less than the degree of f_3. Irreducibility of f_3 ensures irreducibility of g_1 and g_2 over $\mathbb{Q}$.

From Example 6 it is clear that if a Gröbner basis of the polynomial system($\mathcal{GS}$ in our case) can be brought in the form similar to that of (3.9), then we can show

that a limited version of Conjecture 1 is true. Following is generalization of the Example 6.

Proposition 2. *Consider a system of polynomial equations over a field $\mathbb{F}$ of the form:*

$$\begin{aligned} x_1 - h_1(x_n) &= 0 \\ x_2 - h_2(x_n) &= 0 \\ &\dots \\ x_{n-1} - h_{n-1}(x_n) &= 0 \\ h_n(x_n) &= 0. \end{aligned} \tag{3.10}$$

If h_n is irreducible over $\mathbb{F}$ with $0 < \deg h_i < \deg h_n,\ 1 \leq i < n$, then, if $(\alpha_1, \alpha_2, \dots, \alpha_n)$ is a solution, $\alpha_i \notin \mathbb{F}$ for all $i \in \{1, 2, \dots, n\}$.

Proof. We know that degree of $h_n \geq 2$. Let $\alpha_n \notin \mathbb{F}$ be the root of h_n, then h_n is a minimal polynomial for α_n. Now, suppose $\alpha_i \in \mathbb{F}$ for some $i \in \{1, \dots, n-1\}$. Applying division algorithm to h_n and $g(x_n) = \alpha_i - h_i(x_n) \in \mathbb{F}[x_n]$, we get $h_n(x_n) = g(x_n)q(x_n) + r(x_n)$, where either $r(x_n) = 0$ or $\deg r(x_n) < \deg g(x_n) = \deg h_i(x_n) < \deg h_n(x_n)$. Substituting $x_n = \alpha_n$, we get $r(x_n) = 0$ which contradicts minimality of h_n, unless $r = 0$. But if $r = 0$, then $h_n(x_n) = g(x_n)q(x_n)$, contradicting irreducibility of h_n. □

Consider the statement: if $\deg g_i < \deg g_n$ and g_i is irreducible in the system of form (3.10) then g_n is irreducible. The statement can be shown to be incorrect with the following example. Let $x - (y^2 + 1) = 0,\ (y^2 + 1)(y^2 - 2) = 0$. The polynomial $y^2 + 1$ is irreducible over $\mathbb{Q}$, but $y^4 - y^2 - 2$ is not, i.e., statement of the Proposition 2 can not be tightened further. If a polynomial system follows the form(shape) in (3.10), it follows Conjecture 1. The converse is not necessarily true. This makes Proposition 2 only a sufficient condition for Conjecture 1.

The property of the polynomial system in Proposition 2 means that for Conjecture 1 to be true we must bring $\mathcal{GS}$ into form (3.10) and further show that $0 < \deg g_i < \deg g_n,\ 1 \leq i < n$.

3.2.2 Membership Lemma

A Gröbner basis with zero-dimensional ideal $\mathcal{I}$ satisfies the form (3.10) under the following conditions.

Theorem 3.2.3 (Shape Lemma)**.** *[35] Let $\mathcal{I}$ be a zero-dimensional ideal in $\mathbb{F}[x_1, \ldots, x_n]$ which is in general position with respect to x_1, i.e. the projection of $\mathcal{V}_{\mathbb{K}}(\mathcal{I})$ onto the 1-st coordinate is injective. Then $\sqrt{\mathcal{I}}$ has a lexicographic reduced Gröbner basis with respect to $x_n \prec \ldots \prec x_1$ of the form:*

$$\sqrt{\mathcal{I}} = \langle g_n(x_n), x_{n-1} - g_{n-1}(x_n), \ldots, x_2 - g_2(x_n), x_1 - g_1(x_n) \rangle$$

where g_n is a square-free polynomial and the degree of every g_i doesn't exceed degree d of g_n. Here $\mathbb{K}$ denotes algebraic closure of $\mathbb{F}$.

In other words to obtain form (3.10), we need to construct radical of ideal $\mathcal{I}$ of $\mathcal{GS}$, and further show that it is in general position. In case this can be done, we get the following:

Proposition 3. *(Membership Lemma) If the polynomial ideal $\mathcal{I}$ of the game system $\mathcal{GS}$ is zero- dimensional, radical and in general position, then for deciding membership to the class of games exactly one irreducible univariate polynomial in the Gröbner basis of the $\mathcal{GS}$ is sufficient.*

Proof. Result follows from Theorem 3.2.3. □

Worst case computational complexity of the Algorithm 3.1.1 is $\mathcal{O}(\mathcal{K}^+ \cdot 2^{2^{\mathcal{K}^+}})$. In the light of Proposition 3, the for loop in the Algorithm 3.1.1 must run exactly once, consequently the complexity of the Algorithm 3.1.1 is reduced by the factor of $\mathcal{K}^+$, making it $\mathcal{O}(2^{2^{\mathcal{K}^+}})$.

We already know that the ideal $\mathcal{I}$ of the game system $\mathcal{GS}$ that we consider is zero-dimensional. From Theorem 3.2.3 it is clear that $\sqrt{\mathcal{I}}$ of $\mathcal{I}$ can be converted to the form (3.10). This means that we must show that $\mathcal{I}$ is radical and is in general position. Consider the following example.

Example 7.

$$\begin{aligned} x^2 - y &= 0 \\ y - 4 &= 0 \end{aligned} \tag{3.11}$$

and

$$\begin{aligned} x^2 - y + 1 &= 0 \\ y - 1 &= 0 \end{aligned} \tag{3.12}$$

Observe that the system (3.11) has solutions $(2, 4)$ and $(-2, 4)$, and the system (3.12) has solution $(0, 1)$. Further, it is impossible to convert systems (3.11) and (3.12) so that they conform to (3.10) such that $x - f(y)$ generate the roots 2 and -2 for (3.11) and the root 0 with multiplicity 2 for (3.12).

The example above shows the necessity of general position condition.

3.2.4 Radical Ideal: Some Assumed Results

Next we consider various conditions under which the ideal $\mathcal{I}$ is radical.

Elimination Ideal Approach This approach first constructs $\sqrt{\mathcal{I}}$ and further compares it with $\mathcal{I}$ to establish $\mathcal{I} = \sqrt{\mathcal{I}}$.

Theorem 3.2.5. *(SEIDENBERG LEMMA) [56] Let $\mathcal{I} \subset \mathbb{F}[x_1, \ldots, x_n]$ (with $\mathbb{F}$ a perfect field) be a zero-dimensional ideal and $\mathcal{I} \cap \mathbb{F}[x_i] = \langle g_i \rangle$, for $i = 1, \ldots, n$. Let $f_i = \sqrt{g_i} = \frac{g_i}{gcd(g_i, g_i')}$ the square free part of g_i. Then*

$$\sqrt{\mathcal{I}} = \langle \mathcal{I}, f_1, \ldots, f_n \rangle.$$

An alternate and very useful statement of Seidenberg's Lemma from [52] is:

Theorem 3.2.6. *Let $\mathbb{F}$ be a field , let $P = \mathbb{F}[x_1, \ldots, x_n]$, and let $\mathcal{I} \subseteq P$ be a zero-dimensional ideal. Suppose that, for every $i \in \{1, 2, \ldots, n\}$, there exists a non-zero polynomial $g_i \in I \cap \mathbb{F}[x_i]$ such that $gcd(g_i, g_i') = 1$. Then $\mathcal{I}$ is a radical ideal.*

Both, Theorems 3.2.5 and 3.2.6 are constructive in nature. Theorem 3.2.5 utilizes a mechanism to verify $\mathcal{I} = \sqrt{\mathcal{I}}$, while Theorem 3.2.6 does not need any such further verification after the construction. Theorem 3.2.6 is more practical. For computing generator of elimination ideal, [16](p.41) suggests `univpoly` command of `Groebner` package in Maple software.

Next, consider another approach from [83] (Theorem 2.2, p. 14).

Theorem 3.2.7. *A zero-dimensional ideal is radical if and only if the i^{th}-elimination ideals $\mathcal{I} \cap \mathbb{Q}[x_i]$ are radical, $i \in \{1, \ldots, n\}$.*

Theorem 3.2.7 is essentially the same as Theorem 3.2.6. The i^{th} elimination ideal is $\mathcal{I} \cap \mathbb{Q}[x_i] = \langle g_i \rangle$ and $\sqrt{g_i} = f_i = \frac{g_i}{\gcd_{(g_i, g_i')}}$. And when $\gcd(g_i, g_i') = 1$ the ideal $\mathcal{I}$ becomes radical. All the three approaches require computation of elimination ideal for all the indeterminate variables. Complexity of computing radical of an ideal is doubly exponential [56].

Variety Evaluation Approach There are two more approaches for determining whether the polynomial ideal $\mathcal{I}$ is radical. These methods directly work on the $\mathcal{GS}$ and do not require computation of an elimination ideal or a Gröbner basis.

The first approach uses the vanishing ideal for constructing a radical ideal. Hilbert's (strong) Nullstalensatz says that a vanishing ideal $\mathcal{I}_\mathcal{V}$ of a variety set $\mathcal{V}$ over an algebraically closed field $\mathbb{K}$ is radical, i.e., $\mathcal{I}(\mathcal{V}(\mathcal{I}))$ is radical. Considering this we can do the following: first compute solution set (variety) $\mathcal{V}(\mathcal{I})$ of the ideal $\mathcal{I}$ of the $\mathcal{GS}$ and then compute vanishing ideal $\mathcal{I}_\mathcal{V}$ of $\mathcal{V}(\mathcal{I})$. Ideal $\mathcal{I}_\mathcal{V}$ is radical by Hilbert's Nullstalensatz.

The other approach is as follows:

Theorem 3.2.8. *[53] Let $f_1, \ldots, f_n \in \mathbb{F}[x]$ with $\mathbb{F} \in \{\mathbb{R}, \mathbb{Q}\}$ and $x \in \mathbb{C}^n$. Then ideal $\mathcal{I} = \langle f_1, \ldots, f_n \rangle$ is zero-dimensional and radical if for all $x \in \mathcal{V}(\mathcal{I})$, it holds that*

$$det(\partial(f_1(x), \ldots, f_n(x))) \neq 0,$$

where the determinant is that of a Jacobian matrix evaluated at every solution point of the polynomial system.

Methods of Hilbert's Nullstalensatz and Theorem 3.2.8 require computation of solution of the system of polynomial equations, and are not practical for our requirements.

Example 8. The game in Table 3.4 has 3 players and each player has two strategies. The game is originally defined in Nau et al.[70] and is known to have all irrational equilibria.

		A	**B**
	a	3, 0, 2	0, 2, 0
1	**b**	0, 1, 0	1, 0, 0
2	**a**	1, 0, 0	0, 1, 0
	b	0, 3, 0	2, 0, 3

Table 3.4: Payoff table of a 3-player 2-strategy game. Player 1 and 2's strategies are indicated by a, b and A, B respectively. Player 3's strategies are 1 and 2.

We denote first strategies of each player by x, y, z respectively. Then the probability of choosing second strategy is $1-x, 1-y$ and $1-z$. After characterizing Nash equilibria of the game as solutions of a $\mathcal{GS}$ of the form (2.7) we get:

$$\begin{aligned}
&(-1+x)x(-1+y+z+yz) = 0\\
&-(-1+x)x(-1+y+z+yz) = 0\\
&(-1+y)y(3+x(-2+z)-4z) = 0\\
&-(-1+y)y(3+x(-2+z)-4z) = 0\\
&-(3+x(-3+y)-3y)(-1+z)z = 0\\
&(3+x(-3+y)-3y)(-1+z)z = 0.
\end{aligned} \tag{3.13}$$

Applying Buchberger's algorithm with lexicographical order $z \prec x \prec y$, we com-

pute a Gröbner basis as follows:

$$\{3y - 11y^2 + 7y^3 + y^4, 2y - y^2 - y^3 - 5yz + 5y^2z, 4y - 5xy - 5y^2 + 5xy^2 + y^3,$$
$$2y - y^2 - y^3 + 25z - 25xz - 25yz - 25z^2 + 25xz^2 + 25yz^2,$$
$$25x - 25x^2 - 4y - 25xy + 25x^2y + 5y^2 - y^3 - 25xz + 25x^2z\}. \quad (3.14)$$

The polynomial $f = y^4 + 7y^3 - 11y^2 + 3y = 0$ has factors $0, 1, y^2 + 8y - 3$ over $\mathbb{Q}$. Substituting rational roots in the triangular form (3.14) we get $(0, 0, 0)$ and $(1, 1, 1)$ as two solutions of the $\mathcal{GS}$. To verify these solutions for Nash equilibrium we compute payoff of each player at both their pure strategies:

	Probability and Payoff at	
	Strategy 1	Strategy 2
Player 1	0,1	1,2
Player 2	0,3	1,0
Player 3	0,0	1,3

Table 3.5: Strategy payoff table of the game given in Table 3.4. Entry in each cell indicates a probability that the player assigns to his strategy and payoff received for the respective assignment.

Table 3.5 indicates that for Players 2 and 3 their respective payoffs do not maximize for the given choice of strategies. Thus the solution tuple (0,0,0) does not constitute a Nash equilibrium of the game. Similarly, it can verified that the tuple (1,1,1) also does not constitute a Nash equilibrium of the game. What remains are the irreducible factors over $\mathbb{Q}$. This ensures that the game is a member to the class of IPIE games.

Observe that the ideal of the $\mathcal{GS}$ in (3.13) follows Proposition 3.2.2 and so it is sufficient to verify irreducibility of the polynomial in indeterminate y for deciding the membership of the game given in Table 3.4.

As the above discussion and examples indicated, while we have not been able to prove Conjecture 1, it is feasible to establish stronger conditions under which the conclusion would still hold. These conditions can be verified for specific examples, significantly improving the working complexity of our main Membership

Algorithm.

3.3 Discussion

In this chapter we presented an algorithm for deciding membership to the classes of games that we defined in Chapter 2. In the chapters to follow, we present algorithms for computing all Nash equilibria of the member games.

It is important to note that, the method for deciding membership does not assume that the Galois groups are known. Also, for deciding the membership, the algorithm 3.1.1 does not need to compute all the solutions of the $\mathcal{GS}$.

Proposition 3 is particularly important not only to the problem of deciding membership to the classes of games. In next chapter, we shall discuss its importance in problem of computing all equilibria of the classes of games. Moreover, the discussion also suggests a link between the problem of deciding membership and computing equilibria for the class of IPIE and RPIE games.

In this chapter, we presented an important property, Proposition 3, for the ideal of the game system. The property throws more light on the structure of the game system. It would be interesting to see some weaker form of Conjecture 1 to be true for the $\mathcal{GS}$ of any finite normal form game.

Consider an alternate to the algebraic study above, the study of the Membership algorithm under perturbation. We perturb a member RPIE(IPIE) game with independent and identically distributed random number ρ. If the perturbation generates a member game with probability p, then we can consider analyse smoothed complexity of the Algorithms 3.1.1[12]. This complexity analysis provides average running time on the standard inputs.

Chapter 4

Rational Payoff Irrational Equilibria Games

In the previous chapter we presented an algorithm for deciding membership of the classes of games of interest to us. Once it is known that an input game is a member, we proceed to the problem of computing its equilibria. In this chapter we primarily focus on RPIE games and present an algorithm to compute all their equilibria. The correctness of the algorithm and other related results are proved. A detailed example is presented to show the working of the algorithm. We conclude this chapter with a discussion of the complexity of the given algorithm and related issues.

4.1 Underlying Model

RPIE games, by Definition 2.1.4, have totally mixed Nash equilibria. For this reason, we characterize an input RPIE game via a $\mathcal{GS}$ of the form (2.7). Our algorithm for computing all equilibria of the class of RPIE games makes use of Galois groups. The $\mathcal{GS}$ of any RPIE game generates field extensions of $\mathbb{Q}$. Due to this, all the Galois groups in this chapter satisfy Definition 2.3.4.

4.2 Equilibria of RPIE Games

Before formally presenting our algorithm, we briefly discuss the approach and the underlying assumptions. We assume that we have an RPIE game T. As described in Section 2.2, we can derive a system of polynomial equations $\mathcal{GS}$ whose solutions include all Nash equilibria of the game T. However, some of the solutions of the $\mathcal{GS}$ need not be Nash equilibria; our algorithm rejects these unwanted solutions using different mechanisms. From Bernstein's Theorem [4] we have an upper bound on the number of solutions a polynomial system can have. Bounds on the number of equilibria have been given in [63, 64]. These bounds constrain the number of solutions to be computed and the number of non-equilibrium solutions to be rejected.

In the initial phase of our method, Buchberger's algorithm is invoked to derive a univariate polynomial in the Gröbner basis of the $\mathcal{GS}$. [1][2] Since the game is RPIE, Nash's theorem [68] guarantees that the univariate polynomial has at least one irrational root. A root of the univariate polynomial is computed and substituted in the triangular form of a Gröbner basis to find a univariate polynomial in some other indeterminate variable. We repeat this procedure at most $\mathcal{K}^+ - n$ times, and at the end have an irrational solution of the $\mathcal{GS}$, a sample solution.

We denote the Galois group of the irreducible part of a univariate polynomials g_i in the Gröbner basis of $\mathcal{GS}$ by G_i, $i \in \{1, \ldots, \mathcal{K}^+\}$. By assumption G_i's are known.

In the next phase, we apply the transitive Galois group action corresponding to each indeterminate variable and determine all irrational solutions of the $\mathcal{GS}$, from the orbits of the group action.

The final phase consists of testing all the irrational solutions and rejecting the non-equilibrium solutions. For this we invoke the Nash equilibrium verification algorithm in [30]. An outline of the entire algorithm is presented below:

[1] For further details of Buchberger's Algorithm see Section A.2 of Appendix A.

[2] Recall that the $\mathcal{GS}$ has finitely many solutions over the complex number field. This finite variety of the $\mathcal{GS}$ (or equivalently zero-dimensional ideal $\mathcal{I}$ of the $\mathcal{GS}$) guarantees a univariate polynomial in its Gröbner basis [8].

Algorithm 4.2.1 Computing All Nash Equilibria of an RPIE game.
Input: An RPIE game, Galois groups.
Output: All equilibria of the input RPIE game in set X.

1: $\beta = (\beta_1, \beta_2, \ldots, \beta_{\mathcal{K}^+})$. {Initialize an empty tuple to store a sample solution of the $\mathcal{GS}$}.
2: Construct the $\mathcal{GS}$ of the input game.
3: Call Algorithm 4.2.2 with $\mathcal{GS}$ for computing a sample equilibrium of the input RPIE game.
4: Call the Galois group action Algorithm 4.2.3 with the sample solution tuple saved in β.
5: Save output of the Algorithm 4.2.3 in X.
6: Reject non-equilibrium solutions of the $\mathcal{GS}$ from X using verification algorithm in [30] or criteria (2.6) and (2.2).

Algorithm 4.2.2 Computation of a sample solution with Gröbner basis.
Input: $\mathcal{GS}$ of the input game.
Output: A sample solution $\beta = (\beta_1, \beta_2, \ldots, \beta_{\mathcal{K}^+})$ of the input game.

1: With Buchberger's Algorithm on $\mathcal{GS}$, compute triangular form of Gröbner basis.
2: **while** one sample solution β of the $\mathcal{GS}$ is not constructed **do**
3: Compute a root α of univariate polynomial – of some indeterminate variable x_i – generated in Step 3.
4: **if** $\alpha \in \mathbb{Q}$ **then**
5: Reject α and go to Step 3.
6: **else**
7: Save α in β at location β_i.
8: **end if**
9: Substitute the root β_i in β into $\mathcal{GS}$ and compute a new triangular form with one less indeterminate variable.
10: **end while**

Following algorithm computes group action by transitive Galois groups. The action is computed for each indeterminate variable x_i by considering it over each coordinate root in the tuple β. The action generates Galois-conjugates of the roots that are further saved in as solution tuples in the set X.

Algorithm 4.2.3 Computing orbit of a Galois Group Action.
Input: A sample solution β of the $\mathcal{GS}$, Galois groups.
Output: All the conjugate solutions of the input sample solution in set X.

1: Initialize the processed-elements list X and unprocessed-elements list U as $X = U = \{\beta\}$.
2: **while** U is not empty **do**
3: Let $u = (u_1, u_2, \ldots, u_{\mathcal{K}^+})$ be the first element of U. Delete u from U.
4: **for** each i and j, g_j^i in Galois group G_i and $u_i \in u$. **do**
5: Compute the transitive Galois group action $u_i^{g_j^i}$.
6: $\beta' = (u_1^{g_j^1}, u_2^{g_j^2}, \ldots, u_{\mathcal{K}^+}^{g_j^{\mathcal{K}^+}})$.
7: **if** $\beta' \notin X$ **then**
8: $X = X \cup \{\beta'\}$ and $U = U \cup \{\beta'\}$.
9: **end if**
10: **end for**
11: **end while**

Traditional approach, given in [21], for computing solutions of system of polynomial equations using Gröbner basis calls the Buchberger's algorithm for computing a triangular form. The triangular form provides a univariate polynomial in one indeterminate variable. Each root of the univariate polynomial is then substituted back in the triangular form to compute corresponding solution tuple; the operation requires multiple substitutions and factorizations. Algorithm 4.2.1, on the other hand, invokes Buchberger's algorithm exactly once. The Algorithm 4.2.2 computes a sample solution tuple corresponding to the first irrational root of the univariate polynomial. Rest of the solutions are then generated by polynomial time group action, requiring no further substitutions and factorizations.

It is important to note that due to Theorem 2.1.3 and the fact that the input game has all irrational equilibria, we are guaranteed to get one solution of the $\mathcal{GS}$ in β and so the Algorithm 4.2.1 reaches Step 4, every time. It then calls the Algorithm 4.2.3 for computing polynomial time Galois group action over available sample solution in the β. In the Algorithm 4.2.3 all other conjugate roots are computed with their known Galois groups G_i.

Moreover, finite group action on finite variety guarantees that the Algorithm 4.2.3

reaches Step 11. At the end of Step 11, Algorithm 4.2.3 generates solutions of polynomial system $\mathcal{GS}$ in X, all of which may not be Nash equilibria. We use polynomial time algorithm, suggested in [30], to reject the non-equilibrium solutions.

Note that a rational root forces its univariate polynomial to factorize over the field $\mathbb{Q}$. For assuming known Galois group we ignore the linear factor corresponding to each rational root and consider only the irreducible part of the univariate polynomial. We assume that the Galois groups of these irreducible parts are completely known. It also follows from Theorem 2.3.7 that the known Galois groups act transitively on the roots.

4.2.1 Rational Number Check

Details of the condition in Step 4, of Algorithm 4.2.2, for deciding $\alpha \in \mathbb{Q}$ are as follows.

Since numbers are stored in computer memory with finite precision, it is a non-trivial task to determine whether a stored number is rational or irrational. The approach that we have adopted for the problem is as follows.

As a first step, a suitable numerical algorithm (polynomial time root approximation methods), is used to compute an approximate root of the univariate polynomial. Given the approximate root, the degree of the univariate polynomial and its height(defined as the Euclidean length of coefficients of the polynomial), it is possible to construct the minimal polynomial for a root. Finally, checking the irreducibility of the minimal polynomial resolves the problem. The minimal polynomial is constructed using the Kannan Lovasz Lenstra(KLL) algorithm [44]. The irreducibility check can be performed using univariate polynomial factorization algorithm over the field of rational numbers [31]. For further details of the KLL algorithm, see Section A.3.

Algorithm 4.2.2 computes a sample solution of the $\mathcal{GS}$. Various approaches for

computing a sample equilibrium of a game are discussed in [62]. Our approach which makes use of a suitable Gröbner basis of the $\mathcal{GS}$, though not new, is developed independently. Our approach, given in Algorithm 4.2.2, differs from other approaches based on the Gröbner basis in that it focuses on irrational solution tuples of the $\mathcal{GS}$ and ignores its rational solutions.

Algorithm 4.2.1 deploys a method for computing solutions of a system of polynomial equations without having to factorize the system every time.

4.2.2 Results

Algorithm 4.2.1 computes all equilibria of RPIE games with $n \geq 3$ players. We initially show why it does not apply to games with $n = 2$ players.

Proposition 4. *A bimatrix game with all rational payoff values has all rational equilibria.*

Proof. The $\mathcal{GS}$ of a bimatrix game is a system of linear equations [57]. Hence, if all the game payoff values are defined over field $\mathbb{F}$, then all of its solutions can be found in the field $\mathbb{F}$. □

Following is an immediate corollary to the result above.

Corollary 1. *The class of RPIE games is empty for $n = 2$ players.*

Proof. Follows from Definition 2.3.5 and Proposition 4. □

The main result of this section is Proposition 5, which proves the correctness of Algorithm 4.2.1.

Proposition 5. *Algorithm 4.2.1 for computing all equilibria of RPIE games works. i.e., the output at termination consists of all irrational equilibria of the game, and no other solutions of the $\mathcal{GS}$.*

Proof. An input RPIE game T with $n \geq 3$ players is characterized via a $\mathcal{GS}$ of the form (2.7), which is derived from the inequality on expected payoffs and payoffs at pure strategies. Hence, the $\mathcal{GS}$ in general has more solutions than just the

equilibria.

In the first phase, Algorithm 4.2.1 calls Algorithm 4.2.2. Algorithm 4.2.2 computes a sample solution β by first building a Gröbner basis for the $\mathcal{GS}$ using Buchberger's algorithm. Buchberger's algorithm terminates in a triangular form analogous to echelon form in the linear case.

Since the game is known to be RPIE and rational solutions of the $\mathcal{GS}$ are rejected by the Algorithm 4.2.2, the sample solution β must have all irrational coordinates. Consequently, each coordinate β_i of the sample solution β results in an algebraic extension $K = \mathbb{Q}(\beta_i)$ of $\mathbb{Q}$ with finite Galois group $G_i = Gal(\mathbb{K}/\mathbb{Q})$. Since the group action of G_i is transitive, it generates all irrational solutions of the $\mathcal{GS}$.

We know that the $\mathcal{GS}$ has zero-dimensional ideal. This means $\mathcal{GS}$ has finitely many solutions. Group action by a finite Galois group over finite solution set terminates. This enables Algorithm 4.2.1 to reach Step 5 every time there is an RPIE game T as input. The algorithm generates solutions of the $\mathcal{GS}$ that contain all the equilibria of the game T.

Finally, Algorithm 4.2.1 rejects solutions of the $\mathcal{GS}$ which are not Nash equilibria. Since the set of Nash equilibria is known to be non-empty, set X contains all and only the Nash equilibrium solutions of the RPIE game T. □

Note that, Buchberger's algorithm in first phase of the Algorithm 4.2.2 could be replaced by a numerical method to compute a sample equilibrium.

Recall that in Chapter 1 we mentioned the issue of storing irrational equilibria in computer memory, which has been addressed by Lipton and Markakis [60]. The following result shows that the issue can be resolved for a subclass of RPIE games.

Proposition 6. *If univariate polynomials in the Gröbner basis of an RPIE game have solvable Galois groups, then Algorithm 4.2.1 computes Nash equilibria of the game in closed form.*

Proof. It is a standard result that a polynomial with solvable Galois group is solvable by radicals. If each univariate polynomial in the Gröbner basis of an RPIE game has solvable Galois group,[3] then the roots of this set of polynomials can be computed using radicals. This gives all solutions in closed form, which contains set of Nash equilibria of the game. □

It is known that all abelian groups, groups of order < 60, groups of odd order (Feit-Thompson Theorem) and groups of order $p^a q^b$, where p and q are prime, are solvable. Moreover, some non-abelian groups are also be solvable [74]. This suggests that Proposition 6 is applicable to a substantial number of games. The equilibria of games with non-solvable Galois groups can be obtained in algebraic form by first computing equilibrium solutions numerically, and then constructing the minimal polynomials of each of the numerical values with the algorithm in [44].

Recall that in Section 3.2.4, of Chapter 3, we mentioned the importance of Proposition 3 for the problem of computing equilibria. In step 9 of Algorithm 4.2.2 we substitute an irrational root β_i in the triangular form of the Gröbner basis of the $\mathcal{GS}$. If the substitution produces a solution tuple β with mixed coordinates(rational and irrational roots), then we must ignore it. In case Conjecture 1 in Chapter 3 is true for the $\mathcal{GS}$ of the input RPIE game, then such rejection and re-computation of solution tuples can be avoided to increase over all efficiency of the Algorithm 4.2.1.

4.3 Computational Complexity

Constructing of the $\mathcal{GS}$ (Step 2 of Algorithm 4.2.1) is polynomial time in the size of the input payoff matrix, i.e. polynomial time in $\mathcal{K}^*$.

A Gröbner basis can be computed in doubly exponential time in the size of $\mathcal{K}^+$. A Gröbner basis contains polynomials in triangular form and we are interested in the equilibria points with irrational values. An advantage of the triangular form is

[3]It can be verified whether a given polynomial has solvable Galois group or not using the polynomial time Landau-Miller test[55] .

that at every stage of the substitution unwanted solutions can be filtered out. The efficiency of Algorithm 4.2.1 could be significantly improved by replacing Buchberger's algorithm with a more efficient method for computing a sample solution. For further details of computational complexity of finding Gröbner basis, refer Burgisser and Lotz [9].

We are not considering the issue of computing the Galois groups in this work, i.e. we consider that the Galois groups are known. But to make the discussion complete, we give below the complexity of computing the Galois group of a given polynomial. Computation of a Galois group requires polynomial time in the degree of the input polynomial and the order of its Galois group. If $f(x)$ is of degree d then its Galois group can have at most $d!$ elements and so in worst case the computation takes exponential time. This is at present best-known upper bound due to Landau [54]. Lenstra [59] surveys results relating to the complexity of computing Galois groups and other related problems.

Once a Galois group G is known, we must find the Galois orbit $G\beta_i$ of every known root β_i of every indeterminate variable in the $\mathcal{GS}$. An orbit construction takes polynomial time with the algorithm suggested by Luks [61]. In the worst case, the algorithm requires action of each of the Galois group generator $g' \in G' \subseteq G$ to each element of the set of roots. This gives worst case time $O(|G'| \cdot |X|)$. If a univariate polynomial has n roots, then $|G'|$ is linear in n [54, 59], while $|X|$ is polynomial in n. Finally, the verification of a Nash equilibrium solution is a polynomial time operation in the size of total number of strategies $\mathcal{K}^+$.

The algorithm for computing Nash equilibria via the Gröbner basis, given in [21], substitutes all the roots in the triangular form and solves univariate polynomial for each substitution. If the Galois group is known for the polynomials, then our approach computes solutions with relatively simple and efficient group action. Our algorithm exploits information available in a sample solution, and performs better than algorithm which uses only Gröbner basis. If each univariate in the Gröbner basis of the $\mathcal{GS}$ has d_i distinct roots ($i \in \{1, \ldots, \mathcal{K}^+\}$), then the method for computing Nash equilibria in [21] takes $\Pi_i d_i$ substitutions and factorizations.

On the other hand, in our approach, after computing a sample solution, no further substitution or factorization is required.

4.4 Equilibria Computation of an RPIE Game: An Example

In this section, we show working of Algorithm 4.2.1 with an example of 3 players 2 strategy RPIE game. With the Membership Algorithm 3.1.1, given in Chapter 3, we verify that the game, given in Table 4.1 is an RPIE game.

	A	B
a	6, -1, 4	0, 9, 0
b	0, 3/2, 0	2, 0, 0

1

	A	B
a	2, 0, 0	0, 9/2, 0
b	0, 27/2, 0	4, 0, 6

2

Table 4.1: Payoff table of a 3-player 2-strategy RPIE game. Player 1 and 2's strategies are indicated by a, b and A, B respectively. Player 3's strategies are 1 and 2. Entry in each cell of the payoff table indicates player 1, 2 and 3's payoff for their respective strategies.

We let $x = x_1^1, y = x_2^1, z = x_3^1$ be the first strategy of players 1, 2 and 3 respectively. The probability that players will choose their second strategy is $1 - x, 1 - y$ and $1 - z$ respectively. The $\mathcal{GS}$ for the game is as follows:

$$\begin{aligned}
&2(-1+x)(-1+y+z+yz) = 0\\
&2x(-1+y+z+yz) = 0\\
&-2(-1+y)(3+x(-3+z)-3z) = 0\\
&-2y(3+x(-3+z)-3z) = 0\\
&\frac{1}{2}(9-36y+x(9+13y))(-1+z) = 0\\
&\frac{1}{2}(9-36y+x(9+13y))z = 0.
\end{aligned} \tag{4.1}$$

Next, we apply Buchberger's algorithm with lexicographical order $x \prec y \prec z$. The Gröbner basis is:

$$\left\{-27+27x+5x^2, -18-5x+33y, -15+10x+33z\right\}. \tag{4.2}$$

The univariate polynomial $f = -27 + 27x + 5x^2$ has $x = \frac{3}{10}(-9 \pm \sqrt{1}41)$ as its two roots. Hence, f is irreducible over $\mathbb{Q}$ and has Galois group ({id,conjugate}) isomorphic to $\mathbb{Z}_2$.

Substituting $x = \frac{3}{10}(-9 - \sqrt{1}41)$ in the triangular form (4.2) of the Gröbner basis and solving for univariate polynomials in y and z we get: $y = \frac{1}{22}(3 - \sqrt{1}41); z = \frac{1}{11}(14 + \sqrt{1}41)$, a sample solution. The Galois groups of the irreducible polynomials of the $\mathcal{GS}$ are known a priori (isomorphic to $\mathbb{Z}_2$ for each variable x, y, z over $\mathbb{Q}$). All the remaining solutions can be obtained by computing Galois-orbits of the sample solution. The Galois orbits are as follows:

$$\begin{aligned} Gx &= \{\frac{3}{10}(-9 - \sqrt{1}41), \frac{3}{10}(-9 + \sqrt{1}41)\} \\ Gy &= \{\frac{1}{22}(3 - \sqrt{1}41), \frac{1}{22}(3 + \sqrt{1}41)\} \\ Gz &= \{\frac{1}{11}(14 + \sqrt{1}41), \frac{1}{11}(14 - \sqrt{1}41)\}. \end{aligned} \tag{4.3}$$

In this example, it is sufficient to apply the criteria (2.6) and (2.2) for deciding whether a solution tuple is also an equilibrium solution.[4] Accepting values between 0 and 1, we get $x = \frac{3}{10}(-9 + \sqrt{1}41); y = \frac{1}{22}(3 + \sqrt{1}41); z = \frac{1}{11}(14 - \sqrt{1}41)$, as the unique Nash equilibrium of the RPIE game depicted in Table 4.1. Note that the Galois group for the $\mathcal{GS}$ is solvable, and so, all the equilibria computed are in closed form.

4.5 Discussion

In this chapter we presented an algorithm for computing all equilibria for the class of RPIE games which uses Buchberger's algorithm and Galois group action. Our computational complexity analysis suggests that most of the time is consumed in computing a sample solution. In the next chapter we address this issue and present a major modification in the method for computing a sample solution. We also discuss the issue: can we generalize the algebra presented in this chapter to consider other classes of games ?

[4]For larger systems, the Nash equilibrium verification algorithm [30] comes handy.

Chapter 5

Integer Payoff Irrational Equilibria Games

In this chapter, we present a method for computing all the Nash equilibria of an IPIE game. i.e., a game with integer payoffs and irrational equilibria. It differers from the method presented in previous chapter in two significant ways. Firstly, the polynomial algebra and Galois theory required earlier were over the field of rationals. In this chapter, we need to work with the generalized theory of ring extensions and Galois theory over rings. Secondly, instead of using Buchberger's algorithm for identifying a sample solution, we use a numerical algorithm. We also discuss certain properties of the class of IPIE games. We conclude with an example and complexity analysis of the method.

5.1 Underlying Model

We again work with the game system $\mathcal{GS}$ of the form (2.7) as we did with RPIE games. Note that the coefficients of the $\mathcal{GS}$ now come from the ring of integers $\mathbb{Z}$. The solutions of the $\mathcal{GS}$ induce ring extensions rather then field extensions. Following the Galois theory over rings as given in Chase et al. [10], in this chapter we consider Galois ring extensions similar to those defined in 2.3.5. As was done in Algorithm 4.2.1 for RPIE games, the algorithm for IPIE games rejects unwanted solutions of the $\mathcal{GS}$ using suitable mechanisms.

5.2 Equilibria of IPIE Games

In this section, we present an algorithm for computing all Nash equilibria of IPIE games. The algorithm has two stages: in the first stage, compute a sample solution of the $\mathcal{GS}$. Various methods for computing a sample solution are presented in [62]. In this work we use a version of the Multivariate Newton Raphson Method(MVNRM). In the second stage, apply the group action of the Galois group(s) to produce conjugate solutions of the sample solution. Finally, reject all non-equilibrium solutions from the set of solutions to obtain all the equilibria.

Recall that Nash [69] guarantees existence of at least one mixed strategy equilibrium (irrational equilibrium in our case). This implies that for each indeterminate variable, we are guaranteed to get a polynomial with an irreducible non-linear factor over the base ring $\mathbb{Z}$, since the input is an IPIE game.

Being an iterative procedure which converges to a solution, MVNRM starts with an initial guess. The mixed strategy Nash equilibria (probability tuples) form a subset of the set of solutions of $\mathcal{GS}$. This allows choosing an initial guess of either all 0's or 1's or some value between $(0, 1)$. Appropriate choice of a solution tuple speeds up the convergence rate of MVNRM.

Next we have to convert the approximate solution given by MVNRM to algebraic form. Specifically, we need to confirm that the concerned exact solution has all irrational coordinates. For this purpose, we have to use the KLL algorithm, and consequently the stopping rule of the MVNRM iteration has to be tailored to meet the requirements of the KLL algorithm. The KLL algorithm constructs the minimal polynomial of an algebraic number given an approximation to a desired precision, and hence also determines whether the algebraic number is rational or irrational.

The precise calculations are as follows. Suppose that $y = (y_1, \ldots, y_{\mathcal{K}^+})$ is the exact solution, while $\overline{y}_k = (y_{k1}, \ldots, y_{k\mathcal{K}^+})$ is the approximate solution generated by MVNRM at the k-th iteration; as usual $|y|$ indicates the Euclidean norm of the

tuple y.

The KLL algorithm requires $\mathcal{O}(d_i^2 + d_i \log H_i)$ bits of an approximate root for constructing the minimal polynomial of x_i, $i = 1, \ldots, \mathcal{K}^+$ where d_i is degree of the minimal polynomial of x_i and H_i is magnitude bound of the coefficient of the minimal polynomial of y_i.[1] How to determine d_i and H_i is indicated in the following.

Let g_i be the univariate polynomial of the variable x_i in the reduced Gröbner basis of the $\mathcal{GS}$, obtained with a suitable lexicographic ordering. Then, since y_i is a root of g_i, the minimal polynomial of y_i is a factor of g_i, where

$$\begin{aligned} d_i &\leq \deg g_i \\ \text{and } H_i &\leq M_i \\ &\text{where } M_i \text{ is the maximum magnitude of the coefficients of } g_i \end{aligned} \tag{5.1}$$

Further degree of g_i is bounded by number of solutions of the $\mathcal{GS}$ for which Bernstein [4] provides an upper bound.

Proposition 7. *The MVNRM stage of the algorithm must be iterated until the number of zero bits in the binary representation of $|\overline{y}_{k+1} - \overline{y}_k|$ is bounded above by $\mathcal{O}(d^2 + d \log H)$, where $d = \max d_i$ and $H = \max H_i, i = 1, 2 \ldots, \mathcal{K}^+$.*

Proof. If number of zero bits representing $|\overline{y}_{k+1} - \overline{y}_k|$ is bounded above by $\mathcal{O}(d^2 + d \log H)$, then clearly, $\mathcal{O}(d^2 + d \log H)$ bits of y_i in y are available, since

$$\begin{aligned} |y_i - y_{ki}| &\leq |y - \overline{y}_k| \\ &\leq K|\overline{y}_{k+1} - \overline{y}_k|, \end{aligned} \tag{5.2}$$

where K is a constant that depends on the rate of convergence of MVNRM (which is quadratic). □

[1] From Theorem 1.11 of [44] it is clear that we require $|y - \overline{y}_k| \leq 2^{-(d_i^2 + 3d_i + 4d_i \log_2 H_i)}$ for computing a minimal polynomial. A recent result [75] show a tighter bound on the required precision. A function for computing minimal polynomial of an approximate algebraic number is available in Maple and Mathematica softwares.

At the end of the first stage, the algorithm generates a sample solution of the $\mathcal{GS}$.

With the sample solution available (either in algebraic form or in numerical form), in the next stage of the algorithm, we apply group action by Galois groups G. This stage does not differ from the corresponding stage of Algorithm 4.2.1 and so we make use of the Algorithm 4.2.3, presented in Chapter 4. For IPIE games, the Galois groups are associated with ring extensions over $\mathbb{Z}$, and they generate conjugate solutions of the sample solution of the $\mathcal{GS}$. The group action is transitive and produces a single orbit for each indeterminate variable. Using all the orbits we can determine all the irrational solutions of the $\mathcal{GS}$.

Recall that all the solutions of the $\mathcal{GS}$ need not be Nash equilibria. For rejecting unwanted non-equilibrium solutions, we apply the Nash equilibrium verification algorithm.

Algorithm 5.2.1 Computing All Nash Equilibria of an IPIE game.
Input: An IPIE game, Galois groups.
Output: All equilibria of the input IPIE game in set X.

1: $\beta = (\beta_1, \beta_2, \ldots, \beta_{\mathcal{K}^+})$. {Initialize an empty tuple to store a sample solution of the $\mathcal{GS}$}.
2: Construct the $\mathcal{GS}$ of the input game.
3: Call Algorithm 5.2.2 with $\mathcal{GS}$, d and H for computing a sample equilibrium of the input IPIE game.{d and H can be obtained from inequalities in (5.1).}
4: Call the Galois group action algorithm 4.2.3 with the sample solution tuple saved in β.
5: Save output of the Algorithm 4.2.3 in X.
6: Reject non-equilibrium solutions of the $\mathcal{GS}$ from X using verification algorithm in [30] or criteria (2.6) and (2.2).

Algorithm 5.2.2 Computation of a sample solution with MVNRM.
Input: $\mathcal{GS}$ of the input game, d, H.
Output: A sample solution $\beta = (\beta_1, \beta_2, \ldots, \beta_{\mathcal{K}^+})$ of the input game.

1: **while** one sample solution of the $\mathcal{GS}$ is not constructed **do**
2: Apply MVNRM with a starting solution $\mathcal{K}^+$-tuple $u_0 = (0, 0, \ldots, 0)$.
3: **while** inequality in Proposition 7 holds true. **do**
4: Compute approximate solution $u_k = (u_{k1}, u_{k2}, \ldots, u_{k\mathcal{K}^+})$ of $\mathcal{GS}$.
5: **end while**
6: Apply KLL Algorithm on each u_{ki} and compute its minimal polynomial f_i.
7: **if** f_i has linear factor $(x_i - \alpha_i)$ over $\mathbb{Q}$ **then**
8: Obtain a new polynomial system $\mathcal{GS}'$ after factoring out $(x_i - \alpha_i)$ from $\mathcal{GS}$ and go to Step 2 with new $\mathcal{GS}'$
9: **else**
10: Save the solution tuple in β and return.
11: **end if**
12: **end while**

The following result presents details of Step 8 in the Algorithm 5.2.2.

Proposition 8. *After the factorization of linear factor in Step 8 of Algorithm 5.2.2, the new polynomial system $\mathcal{GS}'$ retains subset of solutions of the $\mathcal{GS}$ which has equilibrium solutions with all irrational coordinates.*

Proof. Let $\mathcal{GS} = \{f_i(x_1, \ldots, x_{\mathcal{K}^+}) \in \mathbb{Z}[x_1, \ldots, x_{\mathcal{K}^+}] \mid i \in \{1, \ldots, \mathcal{K}^+\}\}$. Let the variety $\mathcal{V}$ of the $\mathcal{GS}$ be $\{a_1, \ldots, a_m\}$. For some $a_t = (\alpha_{t1}, \ldots, \alpha_{ti}, \ldots, \alpha_{t\mathcal{K}^+}) \in \mathcal{V}$ and $\alpha_{ti} \in \mathbb{Q}$.

We are interested in irrational roots only and so we must find a way to factor $p_i = (x_i - \alpha_{ti})$ from $\mathcal{GS}$. We perform the required factorization in following ways.

Construct $\mathcal{I} = \langle f_1, \ldots, f_{\mathcal{K}^+} \rangle$, then our aim is to compute quotient $\mathcal{I} : \langle p_i \rangle$. The operation requires computation of a Gröbner basis $\{h_1, \ldots, h_k\}$ of $\mathcal{I} \cap \langle p_i \rangle$; then $\mathcal{GS}' = \{\frac{h_1}{p_i}, \ldots, \frac{h_k}{p_i}\}$.

Alternatively, compute a Gröbner basis, with a suitable lexicographic order, such that we get a univariate polynomial $g_k(x_i)$. Then we follow the usual univariate polynomial factorization $g'_k = \frac{g_k(x_i)}{p_i}$ and replace g_k by g'_k in the Gröbner basis to

consider new system of equations.

If $\mathcal{I}$ and $\mathcal{J}$ are any ideals, then computing intersection of ideals does the following to their variety (Ref. Th. 15, Ch. 4, [16])

$$\mathcal{V}(\mathcal{I} \cap \mathcal{J}) = \mathcal{V}(\mathcal{I}) \cup \mathcal{V}(\mathcal{J}).$$

In other words the variety of $\mathcal{I} \cap \langle p_i \rangle$ remains unchanged. Gröbner basis computation does not change $\mathcal{V}$. The only factorization operation is that of p_i that eliminates $x_i - \alpha_{ti} = 0$ and so the ideal of the new system has a variety that is a subset of the original $\mathcal{V}$. Factorization of solution tuples with α_{ti} which include other irrational/rational coordinates are not affecting the desired set of solutions of the $\mathcal{GS}$ that has solutions with all irrational coordinates. □

The procedure of factoring out a root from $\mathcal{I}$ of $\mathcal{GS}$ must be repeated for all $\alpha_{ti} \in \mathbb{Q}$. Due to existence of a mixed Nash equilibrium and the fact that all equilibria are irrational for the input game, we are guaranteed to get one solution of $\mathcal{GS}$ in β and so Algorithm 5.2.2 reaches Step 12 every time. Due to the finiteness of the group and the variety on which it acts, as discussed in Chapter 4, the Algorithm 4.2.3 terminates and so does the Algorithm 5.2.1.

Note that the above method also computes solutions to a system of polynomial equation using its sample solution and its Galois group. After the computation of a sample solution, all other solutions computed are without factorization of the system of polynomial equations.

5.2.1 Some Properties of IPIE Games

From an algorithmic point of view, the main difference between Algorithm 4.2.1 and 5.2.1 is related to the computation of the sample solution. Algorithm 4.2.2 is a purely algebraic approach relying essentially on Buchberger's Algorithm, whereas Algorithm 5.2.2 uses a numerical technique (MVNRM).

Apart from this, differences arise because the algebraic setting for Algorithm 5.2.1

is that of ring extensions of $\mathbb{Z}$. However, because linear factors over $\mathbb{Q}$ are eliminated in Step 7 the algorithm does not compute rational solutions, i.e., those lying in rational extensions of $\mathbb{Z}$. This is stated more precisely below.

Proposition 9. *Algorithm 5.2.1 does not apply to (compute all equilibria of) the case of a game with integral payoffs and at least one rational (Nash) equilibrium.*

Proof. Let T be a game with integer payoffs and one or more rational equilibria of the form a/b, where $a, b \neq 0 \in \mathbb{Z}$. This forces an extension $\mathbb{K} = \mathbb{Z}(a/b)$ over $\mathbb{Z}$. The group G of automorphisms of $\mathbb{K}$ which fix $\mathbb{Z}$ can be computed as follows.

Let $c, d \in \mathbb{Z}$, for any $c + (a/b)d \in \mathbb{K}$ and for any $\sigma \neq id \in G$,

$$\begin{aligned} \sigma(c + (a/b)d) &= \sigma(c) + \sigma(a/b)\sigma(d) \\ &= c + \sigma(a/b)d, \end{aligned}$$

and $\sigma(\frac{a}{b} \cdot b) = a \Rightarrow \sigma(\frac{a}{b})\sigma(b) = a \Rightarrow \sigma(\frac{a}{b}) = a/b \Rightarrow \sigma =$ identity.
This means, the group of automorphisms of rational extensions of the ring of integers turns out to be a trivial identity group. And so, the group doesn't provide necessary information for producing conjugate solutions of the $\mathcal{GS}$. □

Now, in order to prove the validity of the proposed method for IPIE games, we establish three more results, as follows:

Proposition 10. *For any IPIE game, Galois groups representing its equilibrium solutions are non-trivial.*

Proof. Equilibrium solutions of any IPIE game, by definition, generate irrational ring extensions over the ring of integers $\mathbb{Z}$. Suppose, by the way of contradiction, Galois groups for some of the irrational extensions $\mathbb{Z}(\alpha_i)$ are trivial. i.e., $G_i = Gal(\mathbb{Z}(\alpha_i)/\mathbb{Z}) = \{e\}$. Then the minimal polynomial of each α_i has all its factors linear over $\mathbb{Z}$, and hence $\alpha_i \in \mathbb{Z}$. This is impossible for IPIE games. And so the result follows. □

The next result sets the criteria for the MVNRM to converge to a solution of a $\mathcal{GS}$.

Proposition 11. *Let $x^j = (x_1, x_2, \ldots, x_{\mathcal{K}^+})$ be a j^{th} strategy vector with each x_i denoting a probability of players for strategy i and let $f(x) = (f_1(x), f_2(x), \ldots, f_{\mathcal{K}^+}(x))$, for $f_i \in \mathcal{GS}$. Then MVNRM converges to a sample solution of $\mathcal{GS}$ if the following condition holds: $|f(x)\ J^2(f(x))| < |J(f(x))^2|$.*

Proof. In MVNRM, an approximation of the n^{th} strategy tuple x^n is computed using

$$x^n = x^{n-1} - \frac{f(x^{n-1})}{Jf(x^{n-1})}.$$

If we let

$$\phi(x) = x - \frac{f(x)}{Jf(x)} \tag{5.3}$$

then for overall convergence of MVNRM we need $|\frac{d}{dx}\phi(x)| < 1$. Taking the derivative of (5.3) and simplifying it, we get $|f(x)\ J^2(f(x))| < |J(f(x))^2|$. With this condition, MVNRM converges to a sample solution of the $\mathcal{GS}$. □

With the required tools in hand, we can now show the correctness of the method for computing all Nash equilibria of IPIE games.

Proposition 12. *Algorithm 5.2.1 for computing all equilibria of IPIE games works. i.e., the output at termination consists of all irrational equilibria of the game, and no other solutions of the $\mathcal{GS}$.*

Proof. The input to the Algorithm 5.2.1 is an IPIE game T with n players. All the Nash equilibria of this game are characterized by a polynomial system $\mathcal{GS}$ of the form (2.7). The polynomial system comes from the inequalities on expected payoffs and payoffs at pure strategies. These inequalities cause the system to have more solutions then just equilibria.

Algorithm 5.2.2 computes a sample solution of the $\mathcal{GS}$ using MVNRM and saves it in β. This is justified by the following chain of arguments: Nash [68] guarantees that an equilibrium exists with value in $(0, 1)$. Hence, MVNRM computes an approximate solution to the $\mathcal{GS}$ within the degree of precision determined by Proposition 7. The KLL algorithm determines whether the solution is irrational. If not, then the corresponding rational or integer factor is factored out in Step 7 of Algorithm 5.2.2, and the process is repeated. Since the input game is IPIE,

eventually a sample solution is obtained. Roots in the sample solution extend the ring of integers $\mathbb{Z}$ to some Galois extension $\mathbb{K}$ of it. The Galois correspondence in Chase et al. [10] and irreducible polynomials of univariate polynomials in ideal $\mathcal{I}$ of $\mathcal{GS}$ give meaningful transitive Galois groups $G = Gal(\mathbb{K}/\mathbb{Z})$ for the ring extensions. Proposition 5 ensures that the Algorithm 5.2.1 terminates with all equilibrium solutions of the input IPIE game. □

Since Galois theory covers finite fields as well as arbitrary commutative rings [10], it is natural to ask whether our algorithms can also be extended to these situations. This question can be answered as follows:

Proposition 13. *The algebra and algorithms for IPIE(RPIE) games cannot be extended to work over finite fields and their extensions.*

Proof. If we define a finite normal form game over some finite number field, then the only polynomial algebra that we can consider is congruent-modulo algebra. i.e. polynomial system of form (2.7) will be modulo some prime or prime power. This forces the expected cost function codomain values to be restricted to the finite number field. The payoff functions in games must provide every player a choice over his strategies by suggesting an order between elements in the codomain, where the function maps strategies. It is known that, finite number fields are not ordered fields and so they fail to provide a total order amongst player strategies. Moreover, the available order over finite fields conflict with field operations and we cannot perform polynomial algebra. So, we cannot meaningfully define games, and consider polynomial algebra such as suggested in the Algorithms 5.2.1 and 4.2.1 for computing Nash equilibria of such games. □

Due to Proposition 6 in Chapter 4, we know that for a subclass of RPIE games all its Nash equilibria can be computed in closed form. We now develop an analogous result for a subclass of IPIE games.

It is known that if a polynomial defined over fields has a solvable Galois group, then all its roots can be computed with radicals. If the result generalizes over rings then we can generalize the solvability by radical result, i.e. for some ring S

and a subring R the following holds:

$$R = \mathbb{Z} = L_0 \subset L_1 \subset \ldots \subset L_n = S, \tag{5.4}$$

and $\exists\ \alpha_i \in L_{i+1}$, a natural number n_i, such that $L_{i+1} = L_i(a_i)$ and $\alpha_i^{n_i} \in L_i$, then solvability by radicals can be extended for a subclass of IPIE games. All finite ring extensions need not be radical. With this restriction on the extension of the ring and the definition of Galois theory over rings, we have the following result.

Proposition 14. *If the ring extension associated with an IPIE game is radical, then all the equilibria of the game can be computed in closed form.*

Proof. Follows immediately from the discussion above. □

Note that the numerical approach for computing a sample solution in the Algorithm 5.2.2 can be used to replace the Buchberger's Algorithm in the Chapter 4 for the class of RPIE games. The use of numerical method is not novel but independently developed for the games that we consider. Our method in Algorithm 5.2.2 is general, except a condition in Step 7, and can be considered for other classes of games with necessary modifications.

5.3 Computational Complexity

The characterization of equilibria as solutions to a system of polynomial equation is a polynomial time operation in the size of input payoff matrix, where the size of the matrix is $\mathcal{K}^*$. The while loop in Algorithm 5.2.2 of Steps (1-12) runs until a sample solution of the $\mathcal{GS}$ is computed. For $i \in \{1, \ldots, \mathcal{K}^+\}$ and for each indeterminate variable x_i, let d_i denote the degree of its univariate polynomial in $\mathcal{I}$ of the $\mathcal{GS}$. An irrational root of some indeterminate x_i will be available in at most d_i factorization of its univariate polynomial. This implies that the while loop of steps (1-12) runs for at most $d = \max_i d_i$ times. Average case running time analysis of the Newton's method – for computing approximate roots of a univariate polynomial – is studied by Smale [81, 82]. A sufficient number of the steps for the Newton's method to obtain an approximate zero of a polynomial f, are polynomially bounded by the degree d_i of the polynomial and $1/\rho$, where $\rho \in (0, 1)$

is the probability that the method fails. Kuhn's algorithm improves efficiency by a polynomial factor and provides global convergence. On the other hand Renegar [76] studies the problem of computing approximate solutions of multivariate system of equations using homotopy method and presents an efficient algorithm. Note that these results on the complexity analysis assumes that the numerical method converges.

In the Algorithm 5.2.2, number of operations for constructing a minimal polynomial and checking its irreducibility over $\mathbb{Q}$ are bounded by a polynomial in the size of degree d and maximum norm H of the minimal polynomial [31]. The operation of factoring a solution tuple from multivariate polynomial system in Step 8 of the Algorithm 5.2.2 require computation of ideal quotient, which in turn require computation of Gröbner basis. Step 8 executes only when there is a rational root of some univariate polynomial. The operation takes doubly exponential time in $\mathcal{K}^+$. Keeping aside this time, with these details, we present the following complexity bound for computing a sample solution with the Algorithm 5.2.2.

Proposition 15. *Keeping aside time for operation in Step 8, Algorithm 5.2.2 runs in* $O(\mathcal{K}^+ d(1/\rho + H + dH))$.

Proof. The while loop of (1-12) runs for at most d times. Considering the complexity of computing an approximate root of each univariate polynomial with Newton-Raphson's method, the MVNRM with Proposition 11 runs polynomial in $O(\mathcal{K}^+ d \cdot 1/\rho)$. The KLL Algorithm runs in $O(dH)$, requiring at most $\mathcal{K}^+$ repetition in worst case. The operation of checking irreducibility of a minimal polynomial, in worst case, is required for each indeterminate variable and for every factor of the univariate polynomials. The irreducibility check runs in $O(dH)$. Summing up all these times and rearranging terms we get the result. □

Computational complexity of the group action by Galois group in the Algorithm 4.2.3 is discussed in Section 4.3 of Chapter 4.

5.4 Equilibria Computation of an IPIE Game: An Example

We show the working of Algorithm 5.2.1 by computing all equilibria of a 3-player 2-strategy IPIE game with payoff matrix as in Table 5.1. The Algorithm 3.1.1 for deciding membership to the class of IPIE games confirms the game to be a member to the class of IPIE games.

		A	**B**
	a	3, 0, 2	0, 2, 0
1	**b**	0, 1, 0	1, 0, 0
2	**a**	1, 0, 0	0, 1, 0
	b	0, 3, 0	2, 0, 3

Table 5.1: Payoff matrix of a 3-player 2-strategy IPIE game. Player 1 and 2's strategies are indicated by a, b and A, B respectively. Player 3's strategies are 1 and 2. Entry in each cell of the payoff table indicates player 1, 2 and 3's payoff for their respective strategies.

We denote the probability of choosing first strategies of players 1, 2 and 3 by x, y and z respectively. The probability of choosing second strategies is $1-x, 1-y$ and $1-z$ respectively. First, we characterize Nash equilibria of the game, in Table 5.1, as solutions to the $\mathcal{GS}$ with coefficients from $\mathbb{Z}$.

$$\begin{aligned}
&(-1+x)x(-1+y+z+yz) = 0\\
&-(-1+x)x(-1+y+z+yz) = 0\\
&(-1+y)y(3+x(-2+z)-4z) = 0\\
&-(-1+y)y(3+x(-2+z)-4z) = 0\\
&-(3+x(-3+y)-3y)(-1+z)z = 0\\
&(3+x(-3+y)-3y)(-1+z)z = 0
\end{aligned} \tag{5.5}$$

With $d = 2$, $H = 3$ and the initial guess of a solution tuple consisting of all 0's or 1's, we apply MVNRM and compute the following solution tuple.

$$x := 0.7282202113; \quad y := 0.3588989435; \quad z := 0.4717797888 \tag{5.6}$$

The KLL algorithm over the above solution tuple produces the minimal polynomial of each of the roots.

$$5x^2 - 16x + 9 = 0; \quad y^2 + 8y - 3 = 0; \quad 5z^2 + 4z - 3 = 0. \tag{5.7}$$

These polynomials are irreducible over $\mathbb{Z}$ and their Galois groups are isomorphic to $\mathbb{Z}_2$.To obtain a solution tuple in closed form, we factorize the minimal polynomials.[2] Let one such solution be,

$$x = \frac{1}{5}(8 + \sqrt{19}); y = -4 - \sqrt{19}; z = \frac{1}{5}(-2 - \sqrt{19}). \tag{5.8}$$

This is a sample solution of the $\mathcal{GS}$. Next we perform Galois group action on the sample solution. The action of generating Galois orbits for the solution (5.8) is similar to that given in Example 4.4 of Chapter 4. Once all the solutions are computed, we reject non-equilibria solution of the game with the polynomial time verification algorithm [30]. This gives us the unique irrational equilibrium of the IPIE game.

$$x = \frac{1}{5}(8 - \sqrt{19}); y = -4 + \sqrt{19}; z = \frac{1}{5}(-2 + \sqrt{19}). \tag{5.9}$$

[2]Clearly the Galois groups are solvable.

5.5 Discussion

In this chapter we presented a method for computing all equilibria of an IPIE game. The method in its first phase uses MVNRM and the KLL Algorithm. Newton-Raphson method is an efficient method for computing a sample solution and it does not stop in local minima [62]. This supports our choice of method.

It is particulary important to note that the traditional approaches of computing Nash equilibria with numerical methods produced equilibrium points in approximation form. Algorithm 5.2.1 does not depend on the probability distributions. With the convergence condition in Proposition 11 it is a deterministic method that produces equilibrium solutions in exact form using KLL algorithm.

Chapter 6

Construction of Games

In the previous chapters we discussed methods for computing all Nash equilibria of RPIE and IPIE games. A natural question that follows is: is membership of the class of RPIE(IPIE) games reasonably large?

Given a set $\mathcal{S}$, approaches for constructing games with $\mathcal{S}$ as its equilibrium set are presented in [65, 50, 5, 41]. For a mixed strategy tuple to be a Nash equilibrium tuple [65] presents a necessary and sufficient condition. A similar condition for an arbitrary tuple is presented in [50, 5]. Further generalization of the necessary and sufficient condition for a tuple to be a unique Nash equilibrium for an n-player game is presented in [51]. These results are concerned for a single tuple and guarantees that with the given totally mixed tuple, there exists an n-player game with number of strategies of players: $k_1, k_2 \ldots, k_n$, which follow

$$\max_{i=1,\ldots,n} k_i = k_{i_0} \leq \sum_{i \neq i_0} k_i. \tag{6.1}$$

Existence of more examples of RPIE games with unique Nash equilibrium is confirmed in [70]. In this chapter, our primary interest lies in construction of games with multiple Nash equilibria that satisfy the irrationality criteria given in Definitions 2.1.4 and 2.1.5.

The problem of constructing a finite normal form game with desired properties

is an interesting and important one; it allows designing a game which is likely to result in a desirable outcome. If we can further show that it is possible to generate such games in abundant numbers, then the importance of an algorithm for computing all their Nash equilibria increases. The class of RPIE/IPIE games is a particular instance of this broad problem.

Motivated by the existence result of games, in this chapter we consider various approaches for constructing games with special properties. The underlying game model for the problem of construction of games remains similar to that presented in Section 2.2. In the present chapter, the solution tuples x_i^j of $\mathcal{GS}$ given by (2.5) or (2.7) are known and our objective will be to find unknown coefficients $A^i_{j_1 j_2 \cdots j_n}$ that define the game payoff table.

Note that, to convert an RPIE game into an IPIE game, we can scale the payoff values of RPIE game with least common multiple of the denominators of its coefficients. With this in mind, during rest of the discussion, we consider construction of RPIE games.

6.1 Approaches

Prima facie there are two ways to construct a game: given an equilibrium set $\mathcal{S}$ construct a corresponding game payoff table or, derive conditions on the payoff values and show that under the derived conditions, game with required property exists. In this chapter we discuss the former approach that starts with a given equilibrium set $\mathcal{S}$.

6.1.1 Explicit Construction via Polynomial Ideals

In this approach, we start with a given solution set $\mathcal{S}$ of $\mathcal{GS}$. Recall that our objective is to compute coefficients of the $\mathcal{GS}$. As a first step, we construct a polynomial ideal $\mathcal{I}$ with exactly $\mathcal{S}$ as its solutions. In other words, the variety $\mathcal{V}$ of the $\mathcal{I}$ should be exactly $\mathcal{S}$. A vanishing ideal of the given variety set $\mathcal{V}$, $\mathcal{I}_\mathcal{V} = \{f \in \mathbb{K}[x_1, x_2, \ldots, x_n] \mid f(s) = 0, \forall s \in \mathcal{S}\}$, does the job. Next, we construct

a Gröbner basis $\mathcal{GB}$ of $\mathcal{I}_{\mathcal{V}}$.

Our requirement is that the equilibria solutions of the $\mathcal{GS}$ be $\mathcal{S}$. i.e., for every $f_i \in \mathcal{GS}$, f_i must belong to $\mathcal{I}_{\mathcal{V}}$ computed above. In other words, any $f \in \mathcal{GS}$ is as a linear combination of polynomials $g_j \in \mathcal{GB}$. If r_k denote remainder polynomials after the reduction of each f_i by polynomials in $\mathcal{GB}$, then all r_k's must be zero. But, due to unknown coefficients of the $f_i \in \mathcal{GS}$, r_k's will be polynomials in $A^i_{j_1 j_2 \ldots j_n}$. *Solutions of this system of remainder polynomials constitute the relations $\mathcal{R}_{\mathcal{S}}$ between payoff values of the game with $\mathcal{S}$ as its equilibria solutions.*

Note that, based on the elements of the $\mathcal{S}$, some remainder polynomials may turn out to be zero.[1] This causes system of r_k's to have less equations than unknowns, resulting in dependent solutions and equivalent interrelations $\mathcal{R}_{\mathcal{S}}$ between coefficients or the payoff values of the game. We illustrate the approach by following example.

Example 9. This example shows construction of the 3 players 2 strategy finite normal form game given in Section 5.4. It is known that the game can have at most 2 mixed strategy Nash equilibria [63]. We substitute $n = 3$ and $k_1 = k_2 = k_3 = 2$ in equation (2.7) and form $\mathcal{GS}$ of the game. Bernstein's theorem [4] gives a bound on the number of solutions a $\mathcal{GS}$ can have.

Next we define the desired equilibrium solution set $\mathcal{S}$ for the game.

$$\mathcal{S} = \left\{ \left(\frac{1}{5}(8 - \sqrt{19}),\ -4 + \sqrt{19},\ \frac{1}{5}(-2 + \sqrt{19}) \right), \right. \\ \left. \left(\frac{1}{5}(8 + \sqrt{19}),\ -4 - \sqrt{19},\ \frac{1}{5}(-2 - \sqrt{19}) \right) \right\} \tag{6.2}$$

Note that solution with $\sqrt{19}$ extends $\mathbb{Q}$ and has minimal polynomial of degree 2. This forces us to consider conjugate solution as one of the members of $\mathcal{S}$. Also, in this game all the 3 players have 2 strategies. In that case an element of S is

[1] For example substituting a solution tuple with all zeros in the $\mathcal{GS}$ will result in a zero polynomial system.

typically a 6 tuple. But, due to (2.2) we can write 6 tuple as 3 tuple.[2]

The $\mathcal{I}_{\mathcal{V}}$ of $\mathcal{S}$ is:

$$\mathcal{I}_{\mathcal{V}} = \langle 5x - 6 + 5z,\ 5z^2 + 4z - 3,\ y + 2 - 5z \rangle. \tag{6.3}$$

The Gröbner basis of $\mathcal{I}_{\mathcal{V}}$ with lexicographic order $x \prec y \prec z$ is $\{9-16x+5x^2, -4+5x+y, -6+5x+5z\}$. We construct the remainder polynomials r_k reducing f_i by the Gröbner basis. System (6.4) gives an instance of f_i's of player 1 along with its remainder polynomials. Note that in (6.4) $pijk$,$qijk$ and $rijk$ denote payoff values of players 1, 2 and 3 for their i, j and k strategies respectively.

[2]cf. Elements of $\mathcal{S}$ in (6.6) of Example 10 has similarly 4 tuples instead of 8.

$$
\begin{aligned}
p1 \equiv\ & x * (x * (p111 * y * z + p112 * y * (1-z) + p121 * (1-y) * z + p122 * (1-y) * (1-z)) \\
& + (1-x) * (p211 * y * z + p212 * y * (1-z) + p221 * (1-y) * z + p222 * (1-y) * (1-z)) \\
& - p111 * y * z + p112 * y * (1-z) + p121 * (1-y) * z + p122 * (1-y) * (1-z)) \\
\\
r1 \equiv\ & x * (x * (p111 * y * z + p112 * y * (1-z) + p121 * (1-y) * z + p122 * (1-y) * (1-z)) \\
& + (1-x) * (p211 * y * z + p212 * y * (1-z) + p221 * (1-y) * z + p222 * (1-y) * (1-z)) \\
& - p111 * y * z + p112 * y * (1-z) + p121 * (1-y) * z + p122 * (1-y) * (1-z)) \\
& - ((1/5) * p111 * y * z - (1/5) * p112 * y * z - (1/5) * p121 * z * y + (1/5) * p122 * y * z \\
& - (1/5) * p211 * y * z + (1/5) * p212 * y * z + (1/5) * p221 * z * y - (1/5) * p222 * y * z \\
& - (1/5) * p122 * y + (1/5) * p112 * y + (1/5) * p222 * y - (1/5) * p212 * y + (1/5) * p222 * z \\
& - (1/5) * p221 * z - (1/5) * p122 * z + (1/5) * p121 * z - (1/5) * p222 + (1/5) * p122) * (9 - 16 * x + 5 * x^2) \\
& - ((11/25) * p111 * y * z - (21/25) * p112 * y * z - (21/25) * p121 * z * y + (21/25) * p122 * y * z \\
& - (11/25) * p211 * y * z + (11/25) * p212 * y * z + (11/25) * p221 * z * y - (11/25) * p222 * y * z \\
& + (21/25) * p112 * y - (21/25) * p122 * y - (11/25) * p212 * y + (11/25) * p222 * y - (21/25) * p122 * z \\
& + (21/25) * p121 * z - (11/25) * p221 * z + (11/25) * p222 * z + (21/25) * p122 - (11/25) * p222) * (-4 + 5 * x + y) \\
\\
p2 \equiv\ & (1-x) * (x * (p111 * y * z + p112 * y * (1-z) + p121 * (1-y) * z + p122 * (1-y) * (1-z)) \\
& + (1-x) * (p211 * y * z + p212 * y * (1-z) + p221 * (1-y) * z + p222 * (1-y) * (1-z)) \\
& - p211 * y * z + p212 * y * (1-z) + p221 * (1-y) * z + p222 * (1-y) * (1-z)) \\
\\
r2 \equiv\ & (1-x) * (x * (p111 * y * z + p112 * y * (1-z) + p121 * (1-y) * z + p122 * (1-y) * (1-z)) \\
& + (1-x) * (p211 * y * z + p212 * y * (1-z) + p221 * (1-y) * z + p222 * (1-y) * (1-z)) - p211 * y * z + p212 * y * (1-z) \\
& + p221 * (1-y) * z + p222 * (1-y) * (1-z)) - (-(1/5) * p111 * y * z + (1/5) * p112 * y * z + (1/5) * p121 * z * y \\
& - (1/5) * p122 * y * z + (1/5) * p211 * y * z - (1/5) * p212 * y * z - (1/5) * p221 * z * y + (1/5) * p222 * y * z + (1/5) * p122 * y \\
& - (1/5) * p112 * y - (1/5) * p222 * y + (1/5) * p212 * y - (1/5) * p222 * z + (1/5) * p221 * z + (1/5) * p122 * z - (1/5) * p121 * z \\
& + (1/5) * p222 - (1/5) * p122) * (9 - 16 * x + 5 * x^2) - (-(11/25) * p111 * y * z + (11/25) * p112 * y * z + (11/25) * p121 * z * y \\
& - (11/25) * p122 * y * z + (11/25) * p211 * y * z - (1/25) * p212 * y * z + (1/25) * p221 * z * y + (1/25) * p222 * y * z - (11/25) * p112 * y \\
& + (11/25) * p122 * y + (1/25) * p212 * y - (1/25) * p222 * y + (11/25) * p122 * z - (11/25) * p121 * z + (1/25) * p221 * z \\
& - (1/25) * p222 * z - (11/25) * p122 + (1/25) * p222) * (-4 + 5 * x + y)
\end{aligned}
\tag{6.4}
$$

Substituting x, y, z from $\mathcal{S}$ in the remainder polynomials in (6.4) and solving for the coefficients we get the dependencies $\mathcal{R}_{\mathcal{S}}$:

$$\begin{aligned}
p111 &= -\tfrac{2p121}{3} - 2p122 + p211 + \tfrac{2p221}{3} + 2p222 \\
p112 &= -p121 + p212 + p221 \\
q111 &= q121 - \tfrac{2q212}{3} + \tfrac{2q222}{3} \\
q112 &= q122 - \tfrac{q211}{3} + \tfrac{2q212}{9} + \tfrac{q221}{3} - \tfrac{2q222}{9} \\
r111 &= r112 + \tfrac{2r211}{9} - \tfrac{2r212}{9} - \tfrac{2r221}{3} + \tfrac{2r222}{3} \\
r121 &= r122 - \tfrac{r211}{3} + \tfrac{r212}{3}
\end{aligned} \tag{6.5}$$

We compute several instance solutions of (6.5) and construct game payoff tables. One such instance is given in table 6.1 below.

		A	**B**
1	**a**	13/3, 2/3, 19/9	-1, −287/900, 1
	b	1, 2, 10/3	2, 0, 2
2	**a**	5, 329/100, 0	0, 2, 4
	b	0, 1, 0	2, 0, 3

Table 6.1: Payoff matrix of a 3-player 2-strategy game explicitly constructed with polynomial ideals.

We further apply Algorithm 3.1.1 on each instance of the game, that we compute based on (6.5), and verify its membership to RPIE games. The verification reveals that the games have pure strategy Nash equilibrium solutions apart from unique mixed strategy Nash equilibrium given in $\mathcal{S}$. And so the games are not RPIE games. Note that the game given in Section 5.4 is an instance of (6.5) with unique mixed strategy Nash equilibrium given in $\mathcal{S}$. In other words, the approach of computing a game via polynomial ideals provides a *necessary condition* for the construction of RPIE games, but it is not sufficient.

Proposition 16. *If $\mathcal{GB}$ is the reduced Gröbner basis of $\mathcal{I}_{\mathcal{V}}$ defined by $\mathcal{S}$ for $\mathcal{GS}$ and, then the relations of coefficient $\mathcal{R}_{\mathcal{S}}$, obtained from $\mathcal{GS}$ and $\mathcal{GB}$, is a necessary condition on the payoff values of the game; they are not a sufficient condition.*

Proof. $\mathcal{I}_{\mathcal{V}}$ forms a space of all polynomials having exactly $\mathcal{S}$ as its solutions set. $\mathcal{GB}$ of the $\mathcal{I}_{\mathcal{V}}$ forms the basis of the space. We require $\mathcal{GS}$ to have precisely $\mathcal{S}$ as its solutions and so $f_i \in \mathcal{GS}$ must belong to $\mathcal{I}_{\mathcal{V}}$. And so the coefficients of the game with $\mathcal{S}$ as its only solutions must follow the relation $\mathcal{R}_{\mathcal{S}}$. On the other hand Example 9 provides an instance of the game payoff table that has more equilibria solutions than just $\mathcal{S}$. And so the result follows. □

The polynomial ideal approach provides a definite method for computing instances of games with given set always as a subset of its equilibrium set. It requires computation of a Gröbner basis. Once we compute relations $\mathcal{R}_{\mathcal{S}}$ of payoff entries of type (6.5) then we can compute several instances of the games and verify them for their membership to the desired class of games with appropriate algorithm. The approach is general and can be considered for games that are of the form $\mathcal{GS}$. However, in practice it is difficult to find an instance game that has exactly $\mathcal{S}$ as its equilibrium solutions.

6.1.2 Elementary Symmetric Polynomials

It is clear from the discussion so far that, given a solution set $\mathcal{S}$, for constructing a game we need to relate equilibrium solutions and the payoff values(coefficients) of the $\mathcal{GS}$.

Elementary symmetric polynomials (ESP) give relations between roots of a univariate polynomial and its coefficients. If we are to use this idea for $\mathcal{GS}$, we need an extension of ESP for a multivariate polynomial system. Unfortunately, such an extension, which would imply an extended Galois theory for multivariate polynomials is not available.

Consequently, to follow up this idea, we would have to convert a multivariate polynomial system into a system of univariate polynomials. Using Gröbner basis it is possible to convert multivariate system to univariate polynomials $g_i(x_i) \in \mathcal{GB}$ (obtained by change of lexicographic order for each $i \in \{1, \ldots, \mathcal{K}^+\}$). Note that due to unknown coefficients $A^i_{j_1 j_2 \ldots j_n}$ of $\mathcal{GS}$, $\mathcal{GB}$ must be computed in symbolic form of the unknowns. Further, with knowledge of ESP of a univariate polyno-

mial $g_i(x_i)$ and the desired $\mathcal{S}$, coefficients of $g_i(x_i)$ can be computed.

The problem of establishing coefficient($A^i_{j_1 j_2 \cdots j_n}$) - solution ($\mathcal{S}$) relations using ESP is similar to the problem of establishing coefficient($A^i_{j_1 j_2 \cdots j_n}$) - coefficient ($B^i_{j_1 j_2 \cdots j_n}$) relations between polynomials in $\mathcal{GS}$ and its $\mathcal{GB}$, where $B^i_{j_1 j_2 \cdots j_n}$ denote coefficients of the polynomials in $\mathcal{GB}$. The approach is computationally hard and requires further independent exploration.

6.1.3 Explicit Construction Directly from the $\mathcal{GS}$

This approach of construction of games is relatively simpler than the previous approach based on vanishing ideal. We start with a given equilibrium set $\mathcal{S}$ and substitutes solutions in the $\mathcal{GS}$ directly. The substitution gives a system of linear equations in unknown $A^i_{j_1 j_2 \cdots j_n}$. The linear system of equations will be homogeneous because of the polynomial system $\mathcal{GS}$. Existence of a non-trivial solution of this homogeneous system guarantees infinitely many solutions. If we assume that there is a game with $\mathcal{S}$ as its equilibria then there is at least one non-trivial solution, causing the system to have infinitely many game payoff solutions. Infinitely many game payoff solutions verifies that the equilibria solutions of a game under linear transformations of the payoff values are invariant. In other words, if we define a game ($\mathcal{GS}$) in the space of games – having $\mathcal{GB}$ as its basis – with exactly $\mathcal{S}$ as its equilibria, then all the uniform scalar multiple of the payoffs (coefficients of $\mathcal{GS}$) forms a line in this space. Every game on this line has exactly $\mathcal{S}$ as its equilibria, forming an infinite set of games.

An advantage of this approach is that it is computationally efficient compared to the ideal theoretic approach. Note that the methods of explicit construction of a game based on given equilibria set is general and, with necessary modifications, can be considered for construction of finite normal form games that can be written in the form of a $\mathcal{GS}$.

Example 10. We show explicit construction of a 4-player 2-strategy finite normal form game directly from the $\mathcal{GS}$. The game with 4 players and 2 strategy can have

at most 9 equilibria solution [63]. We start with desired solutions set

$$\mathcal{S} = \left\{ \left(\frac{1}{5}(8-\sqrt{19}),\ -4+\sqrt{19},\ \frac{1}{5}(-2+\sqrt{19}),\ -4+\sqrt{19} \right), \right.$$
$$\left(\frac{1}{5}(8+\sqrt{19}),\ -4-\sqrt{19},\ \frac{1}{5}(-2-\sqrt{19}),\ -4-\sqrt{19} \right)$$
$$\left(-\frac{1}{6}+\frac{\sqrt{19}}{5},\ -\frac{1}{3}+\frac{\sqrt{19}}{6},\ 1-\frac{\sqrt{19}}{5},\ -\frac{1}{3}+\frac{\sqrt{19}}{6} \right),$$
$$\left. \left(-\frac{1}{6}-\frac{\sqrt{19}}{5},\ -\frac{1}{3}-\frac{\sqrt{19}}{6},\ 1+\frac{\sqrt{19}}{5},\ -\frac{1}{3}-\frac{\sqrt{19}}{6} \right) \right\}. \tag{6.6}$$

Substituting $\mathcal{S}$ in $\mathcal{GS}$ with $n = 4$ and $k_i = 2,\ i \in \{1, \ldots, 4\}$ we get system of linear homogeneous equations. We further compute dependencies in (6.7) and solve them for several instances of the solution. One such payoff solution is presented in Table 6.2.

$$\mathcal{R}_{\mathcal{S}} : p1111 = \frac{-5446p1212+486p1222+1275p2111+5446p2212-486p2222}{1275}$$
$$p1112 = -p1211 + \frac{461p1212-876p1222+1275p2112+1275p2211-461p2212+876p2222}{1275}$$
$$p1121 = \frac{1}{425}(-437p1212 - 558p1222 + 425p2121 + 437p2212 + 558p2222)$$
$$p1122 = \frac{1}{425}(337p1212 - 425p1221 - 267p1222 + 425p2122 - 337p2212 + 425p2221 + 267p2222)$$

$$q1111 = \frac{1}{1512405}(1512405q1211 + 2(767553q2111 - 2194011q2112 - 179548q2121 + 770331q2122 - 767553q2211 + 2194011q2212 + 179548q2221 - 770331q2222))$$
$$q1112 = \frac{1}{168045}(168045q1212 - 118239q2111 + 181803q2112 + 35794q2121 - 85818q2122 + 118239q2211 - 181803q2212 - 35794q2221 + 85818q2222)$$
$$q1121 = \frac{1}{504135}(504135q1221 + 186672q2111 - 545409q2112 + 4648q2121 - 78636q2122 - 186672q2211 + 545409q2212 - 4648q2221 + 78636q2222)$$
$$q1122 = \frac{1}{1512405}(1512405q1222 - 412836q2111 + 1144467q2112 - 431969q2121 - 512877q2122 + 412836q2211 - 1144467q2212 + 431969q2221 + 512877q2222)$$

$$r1111 = \frac{1}{8153}(8153r1121 - 6(2219r2112 - 2219r2122 + 2219r2211 - 3023r2212 - 2219r2221 + 3023r2222))$$
$$r1112 = \frac{1}{8153}(8153r1122 - 8153r1211 + 8153r1221 - 23107r2112 + 23107r2122 - 23107r2211 + 42770r2212 + 23107r2221 - 42770r2222)$$
$$r1212 = \frac{1}{8153}(8153r1222 + 9525r2112 - 9525r2122 + 9525r2211 - 20567r2212 - 9525r2221 + 20567r2222)$$
$$r2111 = \frac{1}{8153}(50271r2112 + 8153r2121 - 50271r2122 + 50271r2211 - 103482r2212 - 50271r2221 + 103482r2222)$$

$$s1111 = \frac{1}{1512405}(1512405s1112 + 2(767553s2111 - 767553s2112 - 179548s2121 + 179548s2122 - 2194011s2211 + 2194011s2212 + 770331s2221 - 770331s2222))$$
$$s1121 = \frac{1}{504135}(504135s1122 + 186672s2111 - 186672s2112 + 4648s2121 - 4648s2122 - 545409s2211 + 545409s2212 - 78636s2221 + 78636s2222)$$
$$s1211 = \frac{1}{168045}(168045s1212 - 118239s2111 + 118239s2112 + 35794s2121 - 35794s2122 + 181803s2211 - 181803s2212 - 85818s2221 + 85818s2222)$$
$$s1221 = \frac{1}{1512405}(1512405s1222 - 412836s2111 + 412836s2112 - 431969s2121 + 431969s2122 + 1144467s2211 - 1144467s2212 - 512877s2221 + 512877s2222) \quad (6.7)$$

x_1

			t_1	t_2
		z_1	1247/255, −7850800/302481, 35265/8153, −11273552/504135	284/85, −963280/100827, 3, −1194254/168045
y_1		z_2	4, 2, 0, 659488/56015	1, 1, 3, 3658312/504135
y_2		z_1	1, 1, −199478/8153, 1	1, 2, 2, 2
		z_2	2, 9, 2, 9	6, 4, 1, 4

x_2

			t_1	t_2
		z_1	83/255, 933524/33609, 132279/8153, 2	496/85, 6367832/302481, 4, 3
y_1		z_2	2, 15, −45685/8153, 1	1, 13, 0, 1
y_2		z_1	2, 9, 7, 7	1, 2, 2, 2
		z_2	3, 2, 9, 2	2, 2, 4, 2

Table 6.2: Payoff matrix of a 4-player 2-strategy finite normal form game constructed with linear system of equations.

In (6.7) $pijkl$,$qijkl$,$rijkl$ and $sijk$ denote payoff values of players 1, 2, 3 and 4 for their i, j, k and l strategies respectively. Algorithm 3.1.1 to verify membership of the games computed from (6.7) reveals that, as with the case of ideal theoretic approach, the game has desired set of irrational equilibrium solutions, but additionally there are rational equilibrium solutions. This verifies the claim of Proposition 16.

6.1.4 Perturbation

This approach constructs a new game by perturbing the coefficients of some known games. First we observe some properties of perturbation over a function. Let g be a function that takes coefficients $A^i_{j_1 j_2 \ldots j_n}$ (for the simplicity of usage, in this section we denote coefficients by a_j's) of the $\mathcal{GS}$ to its solutions x_k's. Assuming continuity of g we define the following limit:

$$\lim_{a_j \to a'_j} g(a_j) = x'_k$$

Amount of perturbation ρ_i in the coefficients a_j decides the values of solution x_k, i.e., x'_k can be either rational or irrational number. Based on restriction on the values of coefficients a_j's we have the following possibilities.

Let a_j take some arbitrary value after perturbation with some ρ_i, while we restrict values of x_k to irrational, i.e. x'_k should be irrational. If ρ_i in a_j produce an irrational x'_k, then we are done. Otherwise due to continuity of g, before reaching the limit x'_k, g will encounter an irrational x''_k in the neighbourhood of x'_k. We consider x''_k as the new limit (desired equilibria solution) and stop.

Next, we impose restriction on the values of a_j's. After perturbation ρ_i over a_j, we restrict new a_j (a'_j) to be rational. If the corresponding x'_k is not irrational, then with the continuity of g – while approaching the limit – consider the increment in the root value to be $\frac{x_k + x'_k}{2n}$, for successive $n \in \mathbb{N}$. Through g, the specified increment takes value of x_k to some neighbourhood of an irrational x^0_k. i.e., there exists an irrational x^0_k in an some ϵ-neighbourhood of x'_k. Further it requires to be shown

that corresponding to x_k^0 there is a rational a_j^0 that is in some δ-neighbourhood of a'_j. This existence allow us to consider approximate construction of the game rather than exact.

Note that in the absence of some nice behaving function like g above, we can consider the univariate polynomials in the Gröbner basis of the $\mathcal{GS}$. The following example implements perturbation approach.

Example 11. We show effect of perturbation on the payoff values of the game given by Table 6.1. The payoff values are perturbed by real values $\epsilon_i \in [0, 1]$ – chosen uniformly at random.

After perturbing payoffs in Table 6.1 the modified game payoff table is given by Table 6.3.

		A	**B**
	a	2177/500, 1013/1500, 4799/2250	-124/125, -1397/4500, 503/500
1	**b**	251/250, 501/250, 3023/900	201/100, 3/500, 501/250
2	**a**	501/100, 33/10, 3/500	1/250, 501/50, 1001/250
	b	1/500, 101/100, 1/500	1003/500, 1/500, 301/100

Table 6.3: Payoff matrix of perturbed 3-player 2-strategy finite normal form game given in Table 6.1.

We construct $\mathcal{GS}$ (6.8) for the game and compute its equilibria.

$$
\begin{aligned}
p1 &= -\frac{(-1+x1)(-6+300000y1-299997z1+199987y1z1)}{300000} \\
p2 &= \frac{x1(6+299997z1-y1(300000+199987z1))}{300000} \\
\\
p3 &= -\frac{(-1+y1)(-600003-86997z1+x1(695666+391327z1))}{300000} \\
p4 &= \frac{y1(600003+86997z1-x1(695666+391327z1))}{300000} \\
\\
p5 &= \frac{(37x1(105409+18917y1)-9(300004+99995y1))(-1+z1)}{900000} \\
p6 &= \frac{(37x1(105409+18917y1)-9(300004+99995y1))z1}{900000}
\end{aligned}
\tag{6.8}
$$

Solution of (6.8) retains an irrational equilibrium solution

$$\left\{\frac{9\left(10097608779-\sqrt{31536079145371424737}\right)}{55529128732},\right.$$
$$\frac{-5149093085+\sqrt{31536079145371424737}}{1262874852},$$
$$\left.\frac{-2552101103+\sqrt{31536079145371424737}}{6327734878}\right\}. \tag{6.9}$$

There are other rational equilibria solution of the game that also constitute Nash equilibria of the game.

6.2 Discussion

We presented several approaches for constructing games with special properties. The approaches are general enough to be considered for construction of games that can be characterized as $\mathcal{GS}$. The ideal theoretic approach, though expensive for large games, provides algebraic insight in to the problem. We further present an approach directly based on $\mathcal{GS}$ which is efficient and can be considered for large problems in practice. We presented a necessary condition for the construction of games with desired set as its equilibria. Interesting problem would be to find a sufficient condition and make result in Proposition 16 tight.

Chapter 7

Conclusion

7.1 Closing Remarks

In this work, we addressed the problem of computing Nash equilibria for a subclass of finite normal form games. Considering the complexity of the problem, we presented methods that use information about a single Nash equilibrium to compute all the Nash equilibria. We showed that our methods provide an efficient mechanism for computing equilibria compared to existing methods which use only Gröbner basis or homotopy continuation, given the assumption of known Galois groups.

Algorithm 4.2.2 has an advantage of providing algebraic structure to use polynomial algebra over $\mathcal{GS}$ and further allow computations in algebraic form. We make use of the available algebraic structure while considering the problem of deciding membership to the classes of games. Using the membership lemma, in Proposition 3, we showed that with the given properties of ideal $\mathcal{I}$ of the $\mathcal{GS}$ the membership decision is relatively efficient. We further utilize the algebraic approach in Algorithm 4.2.1 to show that the approach has an added advantage of providing irrational solutions in closed form for a subclass of games.

The MVNRM inherently is not globally convergent but with the convergence criteria in Proposition 11, we ensure that our method converges. We follow up

MVNRM with KLL algorithm to convert approximate solutions into equivalent minimal polynomial form. The algebraic form allows us to decide amount of precision required to compute the solution while solving the minimal polynomials. With our method of converting approximate root to its equivalent algebraic form, we establish the subclass of n player games for which there are exact algorithms for computing all its equilibria, contrary to claim in [45]. We further showed that with our method it is possible to compute irrational equilibria of a subclass of IPIE games in closed form (Proposition 14).

We gave an independent treatment for the subclass of IPIE games using the theory of Galois groups over commutative rings. The distinction of RPIE and IPIE games made it clear that the results for two player games presented in [57] do not go over directly to the class of IPIE games, providing a thread for further exploration.

We use Galois groups for computing equilibria with group action in Algorithm 4.2.3. The assumption of known Galois groups could be relaxed by adjoining an algorithm for computing Galois groups. This would utilize the construction of minimal polynomials using the KLL Algorithm and the Tschirnhaus transformation over minimal polynomials [32].

Finally, we conclude this work by presenting some interesting open issues that throw more light on the structural aspects of $\mathcal{GS}$ and provide insight for the future development of efficient methods for computing Nash equilibria.

7.2 Future Work

We presented several approaches for constructing examples of RPIE and IPIE games. However, the approaches involve some problems which we were unable to address. The primary problem is that we have not been able to find a systematic procedure to prevent the appearance of additional equilibria, and these may turn out to involve rational coordinates.

Another open problem is whether composition of games could be employed to

obtain new RPIE/IPIE games. In other words, under what conditions would the composition of an arbitrary game with an RPIE/IPIE game belong to the same class ?

In Chapter 3, we presented an important property for the ideal $\mathcal{I}$ of the $\mathcal{GS}$ (Conjecture 1 and Proposition 3). It required us to show that the ideal $\mathcal{I}$ is zero-dimensional, in general position and radical. The type of games that we consider are known to have zero-dimensional ideal in general position. Our experiments suggest that the ideal $\mathcal{I}$ is also radical. This leads us to the following:

Conjecture 2. *The polynomial Ideal $\mathcal{I}$ of the game system $\mathcal{GS}$, for a game in class RPIE or IPIE, is radical.*

Conjecture 2 would immediately imply Conjecture 1.

Finally, algebra that we presented in this work relates solutions of the $\mathcal{GS}$ which include Nash equilibria of the game. The relations between set of solution of the $\mathcal{GS}$ required Galois group automorphisms. A much stronger and interesting result would be to establish automorphic relations over the set of Nash equilibria solutions of the $\mathcal{GS}$ of games.

Appendix A

Algorithms

A.1 Symmetry and Structures

Given a square in a plane with marked corners, we want to map one corner of the square to other corners, without lifting or distorting the shape of the square. We can rotate it (clockwise or counter clockwise), flip it (horizontally, vertically or diagonally) or do nothing. If we consider all these actions over a square as a set, then any subsequent application of two actions from the set is again an element of the set. Do-nothing is an action that acts like identity of the set, and for every other action in the set there is a reverse action element inside the set – combining the two produces do-nothing action. The set of such actions over a square forms a group that takes one corner of the square to other, maintaining symmetry of the square.

Definition A.1.1 (Group)**.** A non-empty set G, along with a binary operation $\cdot$ defined over it, is called group if

- for any $a, b \in G$, $a \cdot b \in G$ (closure property).
- for any $a, b, c \in G$, $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ (associative law).
- there exists an element $e \in G$ such that for any $a \in G$, $a \cdot e = e \cdot a = a$ (existence of identity element).
- for every $a \in G$, there exists an element $b \in G$ such that $a \cdot b = b \cdot a = e$ (existence of inverse).

The group of symmetries of a square is called dihedral group D_4 and it has eight action elements in it. The set of integers $\mathbb{Z}$ forms a group with addition operation. It is interesting to see that the set $\mathbb{Z}$ offers more structure if we include multiplication as one more operation with addition. The new structure is called a ring.

Definition A.1.2 (Ring)**.** A non-empty set R with two binary operations $+$ and $\cdot$ is called an associative ring if for all $a, b, c \in R$

- $a + b \in R$.
- $a + b = b + a$.
- $(a + b) + c = a + (b + c)$.
- for every $a \in R$ there is an element $0 \in R$ such that $a + 0 = a$.
- there exists an $-a \in R$ such that $a + (-a) = 0$.
- $a \cdot b \in R$.
- $a \cdot (b \cdot c) = (a \cdot b) \cdot c$.
- $a \cdot (b + c) = a \cdot b + a \cdot c$ and $(b + c) \cdot a = b \cdot a + c \cdot a$ (distributive law between $+$ and $\cdot$).

The set $\mathbb{Z}$ does not form a group under $\cdot$ operation. Definition of group and ring motivates the following definition.

Definition A.1.3 (Field)**.** A field $\mathbb{F}$ is a set with group under both its binary operations $+$ and $\cdot$. A non-empty subset $\mathbb{H}$ of $\mathbb{F}$ is called subfield of $\mathbb{F}$ if it forms field under $\mathbb{F}$'s binary operations $+$ and $\cdot$.

Set of all elements of the form,

$$\mathbb{F}[x] = \{a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n \mid a_i \in \mathbb{F} \text{ for all } i \in \{1, 2, \ldots, n\}\},$$

is called ring of polynomials in the indeterminate x with operations $+$ and $\cdot$ defined as follows.[1]

Let $f(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$ and $g(x) = b_0 + b_1x + b_2x^2 + \ldots + b_mx^m$ in $\mathbb{F}[x]$, then

$$f(x) + g(x) = c_0 + c_1x + c_2x^2 + \ldots + c_kx^k$$

where for each $i, c_i = a_i + b_i$ and

$$f(x) \cdot g(x) = d_0 + d_1x + d_2x^2 + \ldots + d_kx^k$$

where for each $i, d_i = a_i \cdot b_0 + a_{i-1} \cdot b_1 + \ldots + a_0 \cdot b_i$.

If n denotes the degree of polynomial $f(x) \in \mathbb{F}[x]$, then its roots α_i are $f(\alpha_i) = 0, i = \{1, 2, \ldots, n\}$. It is possible that some of the $\alpha_i \notin \mathbb{F}$ and they extend $\mathbb{F}$ to some $\mathbb{K} = \mathbb{F}(\alpha_t)$, $\mathbb{F} \subseteq \mathbb{K}$.

A.2 Gröbner Bases and Buchberger's Algorithm

A Gröbner basis is a generating subset, with special properties, of an ideal $\mathcal{I}$ of a polynomial ring R. Buchberger's algorithm for computing a Gröbner basis of a polynomial ideal $\mathcal{I}$ can be looked upon as a generalization of the Gaussian elimination method for a system of linear equations. Before visiting Burchberger's algorithm we need the following definitions.

Definition A.2.1 (Monomial)**.** For the collection of variables $x_1, \ldots, x_n$, a monomial is a product $x_1^{\alpha_1} \ldots x_n^{\alpha_n}$, where $\alpha_i \in \mathbb{N}$ and $i \in \{1, \ldots, n\}$.

Definition A.2.2 (Total Degree)**.** The total degree of the monomial is defined to be $\sum_{i=1}^{n} \alpha_i$.

Consider x^2yz, then its total degree is $2 + 1 + 1 = 4$. The monomial 1 has total degree zero. If the context of indeterminate variables is clear then the monomial can be alternatively written (2,1,1).

[1] cf. Definition A.2.3.

Definition A.2.3 (Polynomial)**.** Finite linear combinations of monomials with coefficients defined in a base field $\mathbb{F}$ is called a polynomial in variables $x_1, \ldots, x_n$.

Polynomials are of the form,

$$f(x_1, \ldots, x_n) = \sum_i c_i x_1^{\alpha_1^i} x_2^{\alpha_2^i} \ldots x_n^{\alpha_n^i}, \tag{A.1}$$

and the summation in (A.1) is finite over i, $c_i \in \mathbb{F}$. So,

$$f(x, y) = 3x^2 + 4y - i5$$

is a polynomial defined over $\mathbb{C}$. Concept of ideal of a ring is general. Our interest is in ring of polynomials. The following concepts are defined for a polynomial ring.

Definition A.2.4 (Ideal)**.** A non-empty subset $\mathcal{I} \subseteq \mathbb{F}[x_1, \ldots, x_n]$ is said to be an ideal if

- for $f, g \in \mathcal{I}$, $f + g \in \mathcal{I}$ and
- for arbitrary polynomial $p \in \mathbb{F}[x_1, \ldots, x_n]$ and for any polynomial $f \in \mathcal{I}$, $pf \in \mathcal{I}$.

Ideal generated by polynomials $f, g \in \mathbb{F}[x_1, \ldots, x_n]$ is denoted by $\mathcal{I} = \langle f, g \rangle$. For example, $\langle x^2 + 2, 3y^2 + 2x \rangle$ is an ideal defined over $\mathbb{Q}[x, y]$.

Definition A.2.5 (Radical of an Ideal)**.** A non-empty set $\sqrt{I} \subseteq \mathbb{F}[x_1, \ldots, x_n]$ is called radical of an ideal $\mathcal{I}$, if for every $f \in \sqrt{\mathcal{I}}$, $f^n \in \mathcal{I}$ for some $n \geq 1$. Moreover, an ideal $\mathcal{I}$ is said to be radical if $\sqrt{\mathcal{I}} = \mathcal{I}$.

Definition A.2.6 (Prime Ideal)**.** An ideal $\mathcal{I} \subset \mathbb{F}[x_1, x_2, \ldots, x_n]$ is said to be prime if whenever a product $fg \in \mathcal{I}$ then at least $f \in \mathcal{I}$ or $g \in \mathcal{I}$.

Prime ideals are radical.

Definition A.2.7 (Maximal ideal)**.** An ideal $\mathcal{I} \subset \mathbb{F}[x_1, x_2, \ldots, x_n]$ is said to be maximal if there are no ideals $\mathcal{J}$ satisfying $\mathcal{I} \subset \mathcal{J} \subset \mathbb{F}[x_1, x_2, \ldots, x_n]$ other than $\mathcal{J} = \mathcal{I}$ and $\mathcal{J} = \mathbb{F}[x_1, x_2, \ldots, x_n]$.

Radical ideal property says that every radical of an ideal $\sqrt{\mathcal{I}}$ contains ideal $\mathcal{I}$. Following is an important result that defines basis of ideals.

Theorem A.2.8 (Hilbert Basis Theorem)**.** *For every ideal* $\mathcal{I} \subseteq \mathbb{F}[x_1, \ldots, x_n]$ *there is a finite generating set* $\{f_1, f_2, \ldots, f_k\} \subset \mathbb{F}[x_1, \ldots, x_n]$ *such that* $\mathcal{I} = \langle f_1, f_2, \ldots, f_k \rangle$.

Definition A.2.9 (Monomial Order)**.** A monomial order in $\mathbb{F}[x_1, \ldots, x_n]$ is a relation $\prec$ on the set of monomials in $\mathbb{F}[x_1, \ldots, x_n]$ such that

(a) $\prec$ is a total order. i.e. for all monomials α, β, either $\alpha = \beta, \alpha < \beta$ or $\beta < \alpha$.

(b) $\prec$ is compatible with monomial multiplication, i.e. given monomials α, β, if $\alpha \prec \beta$ then for any arbitrary monomial ρ, $\alpha\rho \prec \beta\rho$.

(c) $\prec$ is a well order. i.e. every non-empty subset of monomials has a least element under $\prec$.

For example,

Definition A.2.10 (Lexicographical Order)**.** For the monomials α and β in $\mathbb{F}[x_1, \ldots, x_n]$, we say $\alpha \prec_{lex} \beta$ if left most non-zero entry in $\alpha - \beta$ is positive.

For example, consider the monomial exponents $(0, 3, 0), (1, 1, 2)$ and $(1, 2, 1)$ then with the lexicographical order they are related as: $(1, 2, 1) \prec_{lex} (1, 1, 2) \prec_{lex} (0, 3, 0)$. Another example of lexicographical ordering is that of words listed in dictionary of any language.

Definition A.2.11 (Leading term and Leading coefficient)**.** Leading term of a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ with respect to some monomial order $\prec$ is $c_0 x_1^{\alpha_1^0} x_2^{\alpha_2^0} \ldots x_n^{\alpha_n^0}$, $c_0 \neq 0$, the largest monomial in f. We denote leading term of f under $\prec$ as $LT_{\prec}(f)$. Similarly, the coefficient c_0 of the largest monomial is called leading coefficient of the polynomial f under $\prec$, denoted as $LC_{\prec}(f)$.

For example, $f = 5x^2y^2 + 3x^2yz^2$ has $LT_{\prec_{lex}}(f) = 5x^2y^2$ and $LC_{\prec_{lex}}(f) = 5$, while for graded lexicographical order (an order defined over total degree of monomials) $LT_{\prec_{grevlex}}(f) = 3x^2yz^2$ and $LC_{\prec_{grevlex}}(f) = 3$.

Division Algorithm in $\mathbb{F}[\mathbf{x_1}, \ldots, \mathbf{x_n}]$

Fix any monomial order $\prec$ in $\mathbb{F}[x_1, \ldots, x_n]$, and let $P = (f_1, \ldots, f_k)$ be an ordered k-tuple of polynomials in $\mathbb{F}[x_1, \ldots, x_n]$. Then every $f \in \mathbb{F}[x_1, \ldots, x_n]$ can be written as $f = c_1 f_1 + \ldots + c_k f_k + r$, where, for each i, $c_i, r \in \mathbb{F}[x_1, \ldots, x_n]$ and if $c_i f_i \neq 0$ then $LT_\prec(c_i f_i) \leq LT_\prec(f)$. More so, either $r = 0$ or r is a linear combination of monomials not divisible by $LT_\prec(f_1)$,...,$LT_\prec(f_k)$. The r is called *remainder* of f on division by P, written $r = \overline{f}^P$.

Definition A.2.12 (Gröbner basis)**.** Fix a monomial order $\prec$ on $\mathbb{F}[x_1, \ldots, x_n]$, and let $\mathcal{I} \subseteq \mathbb{F}[x_1, \ldots, x_n]$ be an ideal. A Gröbner basis for $\mathcal{I}$ with respect to $\prec$ is a finite collection of polynomials $G = \{g_1, \ldots, g_m\} \subset \mathcal{I}$ such that for every non-zero polynomial $f \in \mathcal{I}$, $LT_\prec(g_i) \mid LT_\prec(f)$ for some i.

Using Dickson's lemma and Hilbert Basis theorem any ideal $\mathcal{I}$ can be written as $\mathcal{I} = \langle g_1, \ldots, g_m \rangle$.

Definition A.2.13 (S-polynomial)**.** Let $f, g \in \mathbb{F}[x_1, \ldots, x_n]$ be non-zero. Let $LT_\prec(f) = c_0 x_1^{\alpha_1^0} x_2^{\alpha_2^0} \ldots x_n^{\alpha_n^0}$ and $LT_\prec(g) = c_1 x_1^{\alpha_1^1} x_2^{\alpha_2^1} \ldots x_n^{\alpha_n^1}$, for some monomial order $\prec$, $c_0, c_1 \in \mathbb{F}$. If $x_1^{\beta_1} x_2^{\beta_2} \ldots x_n^{\beta_n}$ be least common multiple of $x_1^{\alpha_1^0} x_2^{\alpha_2^0} \ldots x_n^{\alpha_n^0}$ and $x_1^{\alpha_1^1} x_2^{\alpha_2^1} \ldots x_n^{\alpha_n^1}$, then the S-polynomial of f and g, denoted by $S(f, g)$ is,

$$S(f, g) = \frac{x_1^{\beta_1} x_2^{\beta_2} \ldots x_n^{\beta_n}}{LT_\prec(f)} \cdot f - \frac{x_1^{\beta_1} x_2^{\beta_2} \ldots x_n^{\beta_n}}{LT_\prec(g)} \cdot g.$$

A polynomial f reduced modulo a set of polynomials F is denoted by $\overline{f}^F$. We are now ready to give an algorithm for computing Gröbner basis of given set of polynomials.

Algorithm A.2.1 Buchberger's Algorithm for computing Gröbner basis.
Input: Polynomial system $P = (f_1, \ldots, f_k)$.
Output: a Gröbner basis $\mathcal{GB} = \{g_1, \ldots, g_m\}$ for ideal $\mathcal{I} = \langle f_1, \ldots, f_k \rangle$.

1: $G := (f_1, \ldots, f_k)$
2: **repeat**
3: $G' := G$
4: **for** each pair $f \neq g$ in G' **do**
5: $S := \overline{S(f,g)}^{G'}$ {Construct an S-polynomial.}
6: **if** $S \neq 0$ **then**
7: $G := G \cup \{S\}$
8: **end if**
9: **end for**
10: **until** $G = G'$. {Until two sets G and G' are not same.}

The computational complexity of constructing a Gröbner basis for a polynomial ideal is doubly exponential in the number of indeterminate of the polynomial ring. For details on this topic, reader is referred to [15].

A.3 Minimal Polynomial Algorithm

In the Algorithm 4.2.1 of computing equilibria, presented in Chapter 4, Step 6 checks for a computed root for its rationality. The test is for constructing a sample equilibrium. In the problem of deciding membership to the classes of games, given in Chapter 3, we factor a univariate polynomial and check for its rational roots. For checking a number to be rational we make use of the Kannan Lovasz Lenstra(KLL) algorithm [44]. Following are details of the KLL algorithm.

A.3.1 KLL Algorithm

Kannan et al. [44] show that if a complex number α satisfies an irreducible primitive polynomial $p(x)$ of degree d, with integer coefficients, each of magnitude at most H, then given $O(d^2 + d \cdot \log H)$ bits of the binary expansion of the real and complex parts of α, we can find $p(x)$ in deterministic polynomial time. Following are some definitions followed by KLL algorithm.

Definition A.3.2 (Lattice)**.** A lattice in $\mathbb{R}^n$ is a set of the form

$$\left\{ \sum_{i=1}^{k} \lambda_i b_i \ : \ \lambda_i \in \mathbb{Z} \right\}$$

where $b_1, b_2, \ldots, b_k$ are independent vectors in $\mathbb{R}^n$.

The lattice L is said to be generated by the vectors $b_1, b_2, \ldots, b_k$, which form a *basis* of the lattice. The lattice is denoted as $L(b_1, b_2, \ldots, b_k)$. An important result on basis reduction algorithm from Lenstra, Lenstra, Lovasz [58] is as follows.

Theorem A.3.3 (LLL-Basis Reduction)**.** *There is a polynomial time algorithm which on input $b_1, b_2, \ldots, b_k$ independent vectors in $\mathbb{Q}^n$ produces a basis $v_1, v_2, \ldots, v_k$ for $L = L(b_1, b_2, \ldots, b_k)$ such that v_1 has length at most $2^{(k-1)/2} \cdot \Lambda_1(L)$, where $\Lambda_1(L)$ is the length of the shortest non-zero vector in L.*

For the details of this result and basis reduction algorithm, refer [31].

Now, corresponding to every polynomial $f(x) = \sum_{i=0}^{m} a_i x^i \in \mathbb{Z}[x]$ of degree at most m, there is a vector $\overline{f}$ in the lattice L, defined by

$$\overline{f} = \sum_{i=0}^{m} a_i b_i. \tag{A.2}$$

Theorem 1.11 in [44] gives the value of parameter $c = 2^{\frac{3}{2}d^2+2d-1} H^{3d}$ for computing basis reduction algorithm.

Following is outline of the algorithm for computing the minimal polynomial of approximation of an algebraic number α.

Algorithm A.3.1 Computing Minimal Polynomial of an Algebraic Number.
Input: A complex number $\overline{\alpha}$ with real and imaginary parts, bounded above by degree d and height H of algebraic number α that is approximated by $\overline{\alpha}$.
Output: Minimal polynomial of α.

1: **for** $i = 1$ to d **do**
2: Run the basis reduction algorithm on $L(b_0, b_1, \ldots, b_i)$ with the value of c defined above.
3: **if** the first basis vector $\overline{f}$ in the reduced basis satisfies $|\overline{f}|^2 \leq 2^d(2H^2)$ **then**
4: Let $f(x)$ be the polynomial corresponding to $\overline{f}$ by the relation defined in (A.2).
5: **return** the primitive part of $f(x)$ as the minimal polynomial of α.
6: **end**
7: **end if**
8: **end for**

A.4 Multivariate Newton Raphson Method

Multivariate Newton Raphson method (MVNRM) is a generalization of its analogous univariate Newton Raphson(UVNRM) method. We first discuss the UVNRM and show how it generalizes to MVNRM.

For approximating roots of a polynomial $f(x)$, UVNRM successively computes points as follows.

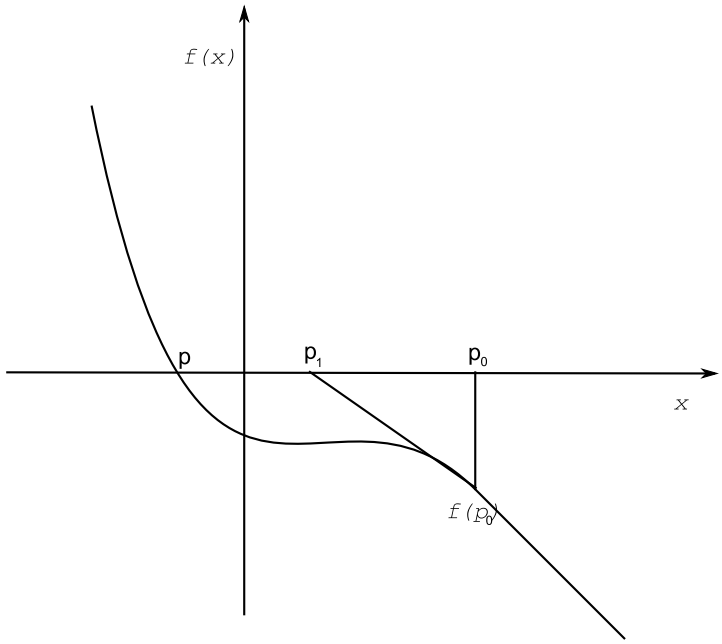

Figure A.1: Approximating root p of polynomial $f(x)$ with Newton-Raphson method

With the initial guess of a root p_0, the idea is to find a linear local region and consider a tangent line(plane in multivariate case) that intersects the indeterminate axis x in a cut point p_1. The point p_1 is the next approximation of the root of $f(x)$. Let the slope of the tangent line be m then,

$$\begin{aligned} m &= \frac{f(p_1) - f(p_0)}{p_1 - p_0} \qquad \text{and} \qquad m = f'(p_0) \\ \therefore p_1 &= p_0 - \frac{f(p_0)}{f'(p_0)} \end{aligned} \tag{A.3}$$

We generalize the equation above and compute p_{i+1}^{th} point with

$$p_{i+1} = p_i - \frac{f(p_i)}{f'(p_i)}. \tag{A.4}$$

Now, let $f : \mathbb{R}^n \to \mathbb{R}^n$ be a vector valued function in several variables. Specifically, for i^{th} tuple $x^i = (x_1^i, x_2^i, \ldots, x_n^i)$, $f(x^i) = (f_1(x^i), f_2(x^i), \ldots, f_n(x^i))$. With this vector valued function, for MVNRM, we replace $f'(p_{i-1})$ in (A.4) by Jacobian

matrix $J(f(x^i))$. The $(i+1)^{th}$ approximate point(in $\mathbb{R}^n$) is computed through

$$x^{i+1} = x^i - J(f(x^i))^{-1} \cdot f(x^i). \tag{A.5}$$

The algorithm for computing approximate solutions of a system of polynomial equations using MVNRM is as follows.

Algorithm A.4.1 Multivariate Newton Raphson Method.
Input: System of polynomial equations $f_1, f_2, \ldots, f_n$, initial guess of solution $(a_1^*, a_2^*, \ldots, a_n^*)$ and threshold τ.
Output: Approximate solution $(\overline{a}_1^0, \overline{a}_2^0, \ldots, \overline{a}_n^0)$.

1: J:= Compute the Jacobian of vector function $f(x)$.
2: Ji:= Compute the inverse of J.
3: $Jifx$:= Multiplication of Matrix Ji with vector $f(x)$.
4: $x^s := (a_1^*, a_2^*, \ldots, a_n^*)$ {Initial guess of the solution tuple}
5: **while** $\epsilon > \tau$ **do**
6: $eJifx$:= evaluate $Jifx$ at x^s.
7: $\epsilon := |x^s - eJifx|$.
8: $x^s := eJifx$.
9: **end while**

The computational complexity of Newton Raphson method is $O(\log n \cdot t)$, where n is the desired number of precision bits of the root value, and t is the time for computing $\frac{f(p_i)}{f'(p_i)}$ in UVNRM, and $J^{-1}(f(x^i))$ in MVNRM. Efficiency of the algorithm can be improved if instead of computing an inverse of the Jacobian matrix we solve the following linear system for x^i and x^{i+1}.

$$J(f(x^i))(x^{i+1} - x^i) = -f(x^i) \tag{A.6}$$

Convergence rate of a system is, roughly, the rate of growth of the error term between subsequent approximations of the root value. Convergence rate of Newton-Raphson(NR) methods in the neighbourhood of a root is quadratic. This means that between two subsequent iterations error is squared, and so the precision is more than doubled. There are quasi-Newton methods for improving the convergence rate of NR methods.

Improper initial guess and vanishing values of the derivatives $f'(p_i)$ or $J^{-1}(f(x^i))$ are the major problems with NR methods. There are other numerical methods to overcome some of these issues. For further details on numerical methods see Chapter 5 of [34].

Appendix B

Implementations

In this appendix we present implementation of Algorithms 3.1.1, 4.2.1 and 5.2.1. Input to all the three algorithms is payoff matrix of the game. In this appendix we confine our selves with 3 players 2 strategy games. For players 1,2 and 3 their respective payoff entries are denoted by p_{ijk}, q_{ijk} and r_{ijk}, where $i, j, k \in \{1, 2\}$ denote pure strategies.

B.1 Membership and Equilibria

The following program first decides membership of the input game by computing a Gröbner basis of it. On affirmative decision of its membership, it further computes its equilibria with group action. The following programs are implemented using Mathematica software.

B.1.1 Computing Gröbner basis

Input

```
Off[General::spell]
p111 = 3
p112 = 0
p121 = 0
p122 = 1
```

```
p211 = 1
p212 = 0
p221 = 0
p222 = 2

q111 = 0
q112 = 1
q211 = 0
q212 = 3
q121 = 2
q122 = 0
q221 = 1
q222 = 0

r111 = 2
r121 = 0
r211 = 0
r221 = 0
r112 = 0
r122 = 0
r212 = 0
r222 = 3

alpha = x*(p111*y*z+p112*y*(1-z)+p121*(1-y)*z+p122*(1-y)*(1-z))
+(1-x)*(p211*y*z+p212*y*(1-z)+p221*(1-y)*z+p222*(1-y)*(1-z))

beta = y*(q111*x*z+q112*x*(1-z)+q211*(1-x)*z+q212*(1-x)*(1-z))
+(1-y)*(q121*x*z+q122*x*(1-z)+q221*(1-x)*z+q222*(1-x)*(1-z))

gemma = z*(r111*x*y+r121*x*(1-y)+r211*(1-x)*y+r221*(1-x)*(1-y))
+(1-z)*(r112*x*y+r122*x*(1-y)+r212*(1-x)*y+r222*(1-x)*(1-y))

(*Above are expected payoff of player 1,2 and 3 respectively.*)
```

```
p1 = x*(alpha-(p111*y*z+p112*y*(1-z)+p121*(1-y)*z+p122*(1-y)*(1-z)))
p2 = (1-x)*(alpha-(p211*y*z+p212*y*(1-z)+p221*(1-y)*z+p222*(1-y)*(1-z)))
p3 = y*(beta-(q111*x*z+q112*x*(1-z)+q211*(1-x)*z+q212*(1-x)*(1-z)))
p4 = (1-y)*(beta-(q121*x*z+q122*x*(1-z)+q221*(1-x)*z+q222*(1-x)*(1-z)))
p5 = z*(gemma-(r111*x*y+r121*x*(1-y)+r211*(1-x)*y+r221*(1-x)*(1-y)))
p6 = (1-z)*(gemma-(r112*x*y+r122*x*(1-y)+r212*(1-x)*y+r222*(1-x)*(1-y)))

(* p1,...,p6 is GS of the input game.  *)

GB = GroebnerBasis[p1, p2, p3, p4, p5, p6, x, z, y]
```

Selected Output

$alpha = (1-x)(2(1-y)(1-z)+yz)+x((1-y)(1-z)+3yz)$
$beta = y(3(1-x)(1-z)+x(1-z))+(1-y)((1-x)z+2xz)$
$gemma = 3(1-x)(1-y)(1-z)+2xyz$

$p1 = (-1+x)x(-1+y+z+yz)$
$p2 = -(-1+x)x(-1+y+z+yz)$
$p3 = (-1+y)y(3+x(-2+z)-4z)$
$p4 = -(-1+y)y(3+x(-2+z)-4z)$
$p5 = -(3+x(-3+y)-3y)(-1+z)z$
$p6 = (3+x(-3+y)-3y)(-1+z)z$

$GB =$

$$\{3y-11y^2+7y^3+y^4, 2y-y^2-y^3-5yz+5y^2z, 4y-5xy-5y^2+5xy^2+y^3,$$
$$2y-y^2-y^3+25z-25xz-25yz-25z^2+25xz^2+25yz^2,$$
$$25x-25x^2-4y-25xy+25x^2y+5y^2-y^3-25xz+25x^2z\}$$

The GB is triangular form of the Gröbner basis of the $\mathcal{GS}$ above. The following piece of code factorizes a univariate polynomial in the triangular form of the GB.

Input

`Factor[`$3y - 11y^2 + 7y^3 + y^4$`]`

Output

$\{(-1 + \mathrm{y})\mathrm{y}\,(-3 + 8\mathrm{y} + \mathrm{y}^2)\}$

For the rational roots $y = 1, 0$ we compute corresponding solution tuple by substituting root values of y in GB. Further we check whether the solutions corresponding to rational values of y constitute Nash equilibria of the input game or not. The solution of $\mathcal{GS}$ corresponding to $y = 0$ is $(x, y, z) = (0, 0, 0)$.

B.1.2 Nash Equilibrium Verification

Input

```
Off[General::spell]
x = 0
y = 0
z = 0

p111 = 3
p112 = 0
p121 = 0
p122 = 1
p211 = 1
p212 = 0
p221 = 0
p222 = 2

q111 = 0
q112 = 1
q211 = 0
q212 = 3
```

```
q121 = 2
q122 = 0
q221 = 1
q222 = 0

r111 = 2
r121 = 0
r211 = 0
r221 = 0
r112 = 0
r122 = 0
r212 = 0
r222 = 3

c11=(p111*y*z+p112*y*(1-z)+p121*(1-y)*z+p122*(1-y)*(1-z))
c12=(p211*y*z+p212*y*(1-z)+p221*(1-y)*z+p222*(1-y)*(1-z))
c21=(q111*x*z+q112*x*(1-z)+q211*(1-x)*z+q212*(1-x)*(1-z))
c22=(q121*x*z+q122*x*(1-z)+q221*(1-x)*z+q222*(1-x)*(1-z))
c31=(r111*x*y+r121*x*(1-y)+r211*(1-x)*y+r221*(1-x)*(1-y))
c32=(r112*x*y+r122*x*(1-y)+r212*(1-x)*y+r222*(1-x)*(1-y))
(* c_ij is payoff of player i when he chooses pure strategy j.*)
```

Output

$c11 = 1$
$c12 = 2$
$c21 = 3$
$c22 = 0$
$c31 = 0$
$c32 = 3$

In terms of Nash equilibrium support, the output above can be interpreted as follows.

- $c11 < c12 \Rightarrow$ For Player 1 the probability of choosing strategy 2 should not be equal to zero. This is true because we started off with $x = 0 \Rightarrow 1 - x = 1$.
- $c21 > c22 \Rightarrow$ For Player 2 the probability of choosing strategy 1 should not be equal to zero. **This is not true because we started off with** $y = 0$**.**
- $c31 < c32 \Rightarrow$ For Player 3 the probability of choosing strategy 2 should not be equal to zero. This is true because we started off with $z = 0 \Rightarrow 1 - z = 1$.

This shows that the tuple $(0, 0, 0)$ is not a Nash equilibrium of the input game. Similar verification for other rational tuple $(x, y, z) = (1, 1, 1)$ produces the same answer. With the help of `IsRadical` function in `PolynomialIdeal` package of Maple software it can be verified that the polynomial ideal $\mathcal{I}$ of the $\mathcal{GS}$ of the input game is radical. This means that the Membership lemma follows for the game, and so it is sufficient to say based on the only irreducible factor of y that the input game is a member game to RPIE(IPIE) games.

B.1.3 Computing Nash equilibrium

Next, we compute Nash equilibria of the game. We solve the irreducible factor $-3 + 8\mathrm{y} + \mathrm{y}^2$ and substitute $y = -4 - \sqrt{19}$ in GB. The new Gröbner basis in indeterminate x and z is

$$\left\{2 + \sqrt{19} + 5z, -8 - \sqrt{19} + 5x\right\}.$$

Over this sample solution we apply group action with known Galois group. The Galois group is of the following form: $G_x = G_y = G_z = \{e, \sigma\}$ where $\sigma(a + \sqrt{19}b) = a - \sqrt{19}b, a, b \in \mathbb{Q}$. The group action on the sample solution produces following orbits of x, y and z.

$$\begin{aligned} Gx &= \{\frac{1}{5}\left(8 + \sqrt{19}\right), \frac{1}{5}\left(8 - \sqrt{19}\right)\} \\ Gy &= \{(-4 - \sqrt{19}), (-4 + \sqrt{19})\} \\ Gz &= \{\frac{1}{5}\left(-2 - \sqrt{19}\right), \frac{1}{5}\left(-2 + \sqrt{19}\right)\} \end{aligned} \tag{B.1}$$

After forming solution tuple with the orbits, we can repeat the Nash equilibrium

verification program in subsection B.1.2 and reject non-equilibrium solution. This lead us to the following unique Nash equilibrium of the input game.

$$x = \frac{1}{5}(8 - \sqrt{19}); y = -4 + \sqrt{19}; z = \frac{1}{5}(-2 + \sqrt{19}) \tag{B.2}$$

B.2 Equilibria of IPIE Game

If the input game is IPIE, then the following program computes its sample solution from its $\mathcal{GS}$ using multivariate Newton-Raphson method(MVNRM). This can be followed by group action. As detailed implementation of group action is given in subsection B.1.3, we shall not reproduce it here. The following program is implemented using Maple software.

B.2.1 Sample Solution with MVNRM

Input

```
with(linalg):
with(Student[MultivariateCalculus]):
with(LinearAlgebra):
p111 := 3:
p112 := 0:
p121 := 0:
p122 := 1:
p211 := 1:
p212 := 0:
p221 := 0:
p222 := 2:

q111 := 0:
q112 := 1:
q211 := 0:
q212 := 3:
q121 := 2:
```

```
q122 := 0:
q221 := 1:
q222 := 0:

r111 := 2:
r121 := 0:
r211 := 0:
r221 := 0:
r112 := 0:
r122 := 0:
r212 := 0:
r222 := 3:

alpha := x1*(p111*y1*z1+p112*y1*z2+p121*y2*z1+p122*y2*z2) +
     x2*(p211*y1*z1+p212*y1*z2+p221*y2*z1+p222*y2*z2):
beta  := y1*(q111*x1*z1+q112*x1*z2+q211*x2*z1+q212*x2*z2) +
     y2*(q121*x1*z1+q122*x1*z2+q221*x2*z1+q222*x2*z2):
gemma := z1*(r111*x1*y1+r121*x1*y2+r211*x2*y1+r221*x2*y2) +
     z2*(r112*x1*y1+r122*x1*y2+r212*x2*y1+r222*x2*y2):
# alpha, beta and gemma are expected payoff of players 1,2 and 3
# respectively.

p1 := (alpha-p111*y1*z1-p112*y1*z2-p121*y2*z1-p122*y2*z2):
p2 := (alpha-p211*y1*z1-p212*y1*z2-p221*y2*z1-p222*y2*z2):
p3 := (beta-q111*x1*z1-q112*x1*z2-q211*x2*z1-q212*x2*z2):
p4 := (beta-q121*x1*z1-q122*x1*z2-q221*x2*z1-q222*x2*z2):
p5 := (gemma-r111*x1*y1-r121*x1*y2-r211*x2*y1-r221*x2*y2):
p6 := (gemma-r112*x1*y1-r122*x1*y2-r212*x2*y1-r222*x2*y2):
# p1,...,p6 constitute game system GS of the input game.

J := Jacobian([p1, p2, p3, p4, p5, p6], [x1, x2, y1, y2, z1,z2]):
# For successive approximation of solution tuple we compute Jacobian
# matrix J.
```

```
Ji := MatrixInverse(J):
# Inverse of the Jacobian matrix J is Ji.

fx := Vector(1..6,[p1, p2, p3, p4, p5, p6]):
# The vector valued function constructed out of the GS.

Jifx:=Ji.fx:
# Evaluation of Ji at fx.



n:=1;

# first guessed solution tuple.
ar := array(1 .. 6,[1/2,1/2,1/2,1/2,1/2,1/2]):
tr := array(1 .. 6,[0, 0, 0, 0, 0, 0]):

while n < 5
do
    eJifx:= eval(Jifx,[x1=ar[1], x2=ar[2],
                y1=ar[3], y2=ar[4], z1=ar[5],z2=ar[6]]):
    tr := eJifx:
    ar[1] := evalf(ar[1]-tr[1]):
    ar[2] := evalf(ar[2]-tr[2]):
    ar[3] := evalf(ar[3]-tr[3]):
    ar[4] := evalf(ar[4]-tr[4]):
    ar[5] := evalf(ar[5]-tr[5]):
    ar[6] := evalf(ar[6]-tr[6]):
    e:= (ar[1]^2+ar[2]^2+ar[3]^2+ar[4]^2+ar[5]^2+ar[6]^2)^(1/2);
     n:=n+1;
end do;
# The while loop above evaluates the successive approximations of
# the sample solution.
```

```
with(PolynomialTools):
ar[1];xm:=MinimalPolynomial(ar[1], 2);
ar[2];MinimalPolynomial(ar[2], 2);
ar[3];ym:=MinimalPolynomial(ar[3], 2);
ar[4];MinimalPolynomial(ar[4], 2);
ar[5];zm:=MinimalPolynomial(ar[5], 2);
ar[6];MinimalPolynomial(ar[6], 2);
# The MinimalPolynomial function converts the approximation of
# roots saved in ar[1],...,ar[6] in to equivalent minimal
# polynomial form.


xroots:=solve(xm);
yroots:=solve(ym);
zroots:=solve(zm);
```

Selected Output

$$n := 1$$

$$eJifx := \begin{bmatrix} -\frac{155}{774} \\ \frac{310}{1161} \\ \frac{181}{1548} \\ -\frac{181}{1161} \\ \frac{16}{387} \\ -\frac{32}{1161} \end{bmatrix}$$

$$tr := \begin{bmatrix} -\frac{155}{774} \\ \frac{310}{1161} \\ \frac{181}{1548} \\ -\frac{181}{1161} \\ \frac{16}{387} \\ -\frac{32}{1161} \end{bmatrix}$$

$$ar_1 := 0.7002583979$$

$$ar_2 := 0.2329888028$$

$$ar_3 := 0.3830749354$$

$$ar_4 := 0.6559000861$$

$$ar_5 := 0.4586563307$$

$$ar_6 := 0.5275624462$$

$$e := 1.268969937$$

$$\vdots$$

$$eJifx := \begin{bmatrix} 0.000004959766902 \\ 0.000003274264997 \\ 0.000003125603050 \\ 0.000006367529572 \\ 0.000001372035271 \\ 0.0000009976326461 \end{bmatrix}$$

$$tr := \begin{bmatrix} 0.000004959766902 \\ 0.000003274264997 \\ 0.000003125603050 \\ 0.000006367529572 \\ 0.000001372035271 \\ 0.0000009976326461 \end{bmatrix}$$

$$ar_1 := 0.7282202113$$
$$ar_2 := 0.2717797887$$
$$ar_3 := 0.3588989435$$
$$ar_4 := 0.6411010566$$
$$ar_5 := 0.4717797888$$
$$ar_6 := 0.5282202113$$
$$e := 1.282801897$$
$$n := 5$$
$$0.7282202113$$
$$xm := 9 - 16\,\_X + 5\,\_X^2$$
$$0.2717797887$$
$$-2 + 6\,\_X + 5\,\_X^2$$
$$0.3588989435$$
$$ym := -3 + 8\,\_X + \_X^2$$
$$0.6411010566$$
$$6 - 10\,\_X + \_X^2$$
$$0.4717797888$$
$$zm := -3 + 4\,\_X + 5\,\_X^2$$
$$0.5282202113$$
$$6 - 14\,\_X + 5\,\_X^2$$
$$xroots := 8/5 + 1/5\,sqrt\,(19)\,,\ 8/5 - 1/5\,sqrt\,(19)$$
$$yroots := -4 + sqrt\,(19)\,,\ -4 - sqrt\,(19)$$
$$zroots := -2/5 + 1/5\,sqrt\,(19)\,,\ -2/5 - 1/5\,sqrt\,(19)$$

References

[1] L. Alvisi, A. Clement, H. Li, F. Mari, I. Melatti, I. Salvo, and E. Tronci, "Model checking coalition Nash equilibria in MAD distributed systems," in *11th International Symposium on Stabilization, Safety, and Security of Distributed Systems*, November 2009.

[2] A. Ash and R. Gross, *Fearless symmetry : Exposing the hidden patterns of numbers*. Princeton University Press, 2006.

[3] I. Bárány, S. Vempala, and A. Vetta, "Nash equilibria in random games," *Random Struct. Algorithms*, vol. 31, no. 4, pp. 391–405, 2007.

[4] D. Bernstein, "The number of roots of a system of equations," *Functional Analysis and its Applications*, vol. 9, pp. 183–185, 1975.

[5] P. Borm and A. Gijsberts, "On constructing games with a convex set of equilibrium strategies," *Mathematical Methods of Operations Research*, vol. 34, no. 4, pp. 279–302, July 1990.

[6] F. Brandt, F. Fischer, and M. Holzer, "Symmetries and the complexity of pure Nash equilibrium," in *Proceedings of the 24th International Symposium on Theoretical Aspects of Computer Science*, ser. LNCS. Aachen, Germany: Springer-Verlag, Feburary 2007.

[7] A. Brodzik, "On the Fourier transform of finite chirps," *IEEE Signal Processing Letters*, vol. 13, no. 9, pp. 541–544, September 2006.

[8] B. Buchberger, "Groebner bases: A short introduction for systems theorists," in *EUROCAST 2001 - 8th International Conference on Computer Aided Systems Theory - Formal Methods and Tools for Computer Science*, ser. Lecture Notes in Computer Science, R. Moreno-Diaz, B. Buchberger, and J. Freire, Eds., vol. 2178. Berlin: Springer-Verlag, 2001, pp. 1 – 19.

[9] P. Buergisser and M. Lotz, "The complexity of computing the Hilbert polynomial of smooth equidimensional complex projective varieties," *Foundations of Computational Mathematics*, vol. 7, no. 1, pp. 51–86, 2007.

[10] S. U. Chase, D. K. Harrison, and A. Rosenberg, "Galois theory and Galois cohomology of commutative rings," *Memoirs of the American Mathematical Society*, no. 52, pp. 15–33, 1965.

[11] X. Chen and X. Deng, "Settling the complexity of two-player Nash equilibrium," in *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 261–272.

[12] X. Chen, X. Deng, and S.-H. Teng, "Computing Nash equilibria: Approximation and smoothed complexity," in *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, 2006.

[13] V. Conitzer and T. Sandholm, "Complexity results about Nash equilibria," in *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, 2003, pp. 765–771.

[14] D. Cox, *Galois Theory*. John Wiley & Sons, 2004.

[15] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, 2nd ed. Springer, November 1996.

[16] ——, *Using Algebraic Geometry*. Springer, 1998.

[17] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, "The complexity of computing a Nash equilibrium," in *STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*. New York, NY, USA: ACM, 2006, pp. 71–78.

[18] ——, "The complexity of computing a Nash equilibrium," *Communications of The ACM*, vol. 52, no. 2, pp. 89–97, 2009.

[19] C. Daskalakis, A. Mehta, and C. Papadimitriou, "A note on approximate Nash equilibria," *Theoretical Compututer Science*, vol. 410, no. 17, pp. 1581–1588, 2009.

[20] C. Daskalakis and C. H. Papadimitriou, "Discretized multinomial distributions and Nash equilibria in anonymous games," in *FOCS '08: Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 25–34.

[21] R. S. Datta, "Finding all Nash equilibria of a finite game using polynomial algebra," *Economic Theory*, vol. 42, no. 1, pp. 55–96, January 2010.

[22] J. Dickhaut and T. R. Kaplan, "A program for finding Nash equilibria," *The Mathematica Journal*, vol. 1, no. 4, pp. 87–93, 1991.

[23] J. Dugundji, *Topology*. Allyn and Bacon Inc., Boston, 1966.

[24] E. Elkind, L. A. Golberg, and P. W. Goldberg, "Computing good Nash equilibria in graphical games," in *EC '07: Proceedings of the 8th ACM conference on Electronic commerce*. New York, NY, USA: ACM, 2007, pp. 162–171.

[25] A. Enge and F. Morain, "Fast decomposition of polynomials with known Galois group," in *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, vol. 2643, 2003, pp. 254–264.

[26] A. Fabrikant, A. Luthra, E. Maneva, C. Papadimitriou, and S. Shenker, "On a network creation game," in *Twenty-Second Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, 2003, pp. 347–351.

[27] A. Fabrikant, C. Papadimitriou, and K. Talwar, "The complexity of pure Nash equilibria," in *STOC '04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*. New York, NY, USA: ACM Press, 2004, pp. 604–612.

[28] D. Fotakis, S. C. Kontogiannis, E. Koutsoupias, M. Mavronicolas, and P. G. Spirakis, "The structure and complexity of Nash equilibria for a selfish routing game," in *ICALP '02: Proceedings of the 29th International Colloquium on Automata, Languages and Programming*. London, UK: Springer-Verlag, 2002, pp. 123–134.

[29] M. Gairing, T. Lcking, M. Mavronicolas, B. Monien, and P. Spirakis, "Extreme Nash equilibria," in *ICTCS '03: Proceedings of the Eighth Italian Conference on Theoretical Computer Science*, ser. LNCS, C. Blundo and C. Laneve, Eds., vol. Volume 2841/2003. Springer Berlin / Heidelberg, 2003, pp. 1–20.

[30] R. Gandhi, "Selfish routing and network creation games," Master's thesis, Dhirubhai Ambani Institute of Information and Com-

munication Technology, Gandhinagar, Gujarat, India, May 2005, http://sites.google.com/site/ratnikg/MTechThesisRGandhi.pdf.

[31] J. v. Z. Gathen and J. Gerhard, *Modern Computer Algebra*. New York, NY, USA: Cambridge University Press, 2003.

[32] K. Geissler and J. Kluners, "Galois group computation for rational polynomials," *Journal of Symbolic Computation*, vol. 11, pp. 1–23, 2000.

[33] I. Gilboa and E. Zemel, "Nash and correlated equilibria: Some complexity considerations," *Games and Economic Behavior*, vol. 1, no. 1, pp. 80–93, March 1989.

[34] A. J. Gomes, I. Voiculescu, J. Jorge, B. Wyvill, and C. Galbraith, *Implicit Curves and Surfaces: Mathematics, Data Structures and Algorithms*. Springer, March 2009.

[35] M.-J. Gonzalez-Lopez and L. Gonzalez-Vega, *Gröbner bases and applications*, ser. London Mathematical Society Lectures Notes Series. Cambridge University Press, 1998, no. 251, ch. Newton identities in the multivariate case: Pham Systems, pp. 351 – 366.

[36] G. Gottlob, G. Greco, and F. Scarcello, "Pure Nash equilibria: hard and easy games," in *TARK '03: Proceedings of the 9th conference on Theoretical aspects of rationality and knowledge*. New York, NY, USA: ACM Press, 2003, pp. 215–230.

[37] S. Govindan and R. Wilson, "A global Newton method to compute Nash equilibria," *Journal of Economic Theory, iss. 1*, vol. 110, pp. 65–86, 2003.

[38] J. C. Harsanyi, "Oddness of the number of equilibrium points: A new proof," *International Journal of Game Theory*, vol. 2, pp. 235–250, 1973.

[39] P. J.-J. Herings and R. Peeters, "A globally convergent algorithm to compute all Nash equilibria for n-person games," *Annals of Operations Research*, vol. 137, no. 1, pp. 349–368, July 2005.

[40] ——, "Homotopy methods to compute equilibria in game theory," *Economic Theory*, vol. 42, no. 1, pp. 1–7, January 2010.

[41] G. A. Heuer, "Uniqueness of equilibrium points in bimatrix games," *International Journal of Game Theory*, vol. 8, no. 1, pp. 13–25, March 1979.

[42] C. U. Jensen, A. Ledet, and N. Yui, *Generic Polynomials : Constructive Aspects of the Inverse Galois Problem*, ser. Mathematical Sciences Research Institute Publications. Cambridge University Press, 2002, no. 45.

[43] Z. J. Ji, W. Yu, and K. J. R. Liu, "An optimal dynamic pricing framework for autonomous mobile ad hoc networks," in *25th IEEE International Conference on Computer Communications*, April 2006, pp. 1–12.

[44] R. Kannan, A. K. Lenstra, and L. Lovasz, "Polynomial factorization and nonrandomness of bits of algebraic and some transcendental numbers," in *STOC '84: Proceedings of the annual ACM symposium on Theory of computing*. ACM, 1984, pp. 191–200.

[45] M. Kearns, M. L. Littman, and S. Singh, "Graphical models for game theory," in *Proceedings of the Seventeenth Annual Conference on Uncertainty in Artificial Intelligence*, 2001, pp. 253–260.

[46] S. U. Khan and I. Ahmad, "A pure Nash equilibrium-based game theoretical method for data replication across multiple servers," *IEEE Transection on Knowledge and Data Engineering*, vol. 21, no. 4, pp. 537–553, 2009.

[47] B. M. Kiernan, "The development of Galois theory from Lagrange to Artin," *Archive for History of Exact Sciences*, vol. 8, no. 1-2, pp. 40–154, January 1971.

[48] D. Koller and N. Megiddo, "Finding mixed strategies with small supports in extensive form games," *International Journal of Game Theory*, vol. 25, no. 1, pp. 73–92, March 1996.

[49] E. Koutsoupias and C. Papadimitriou, "Worst-case equilibria," in *16th Annual Symposium on Theoretical Aspects of Computer Science*, Trier, Germany, 4–6 Mar. 1999, pp. 404–413.

[50] V. L. Kreps, "Bimatrix games with unique equilibrium points," *International Journal of Game Theory*, vol. 3, no. 2, pp. 115–118, 1974.

[51] ——, "Finite n-person non-cooperative games with unique equilibrium points," *International Journal of Game Theory*, vol. 10, no. 3/4, pp. 125–129, 1978.

[52] M. Kreuzer and L. Robbiano, *Computational Commutative Algebra I.* Springer, 2000.

[53] F. Kubler and K. Schmedders, "Competitive equilibria in semi-algebraic economics," *Journal of Economic Theory*, vol. 145, no. 1, pp. 301–330, January 2010.

[54] S. Landau, "Polynomial time algorithms for Galois groups," in *EUROSAM '84: Proceedings of the International Symposium on Symbolic and Algebraic Computation.* London, UK: Springer-Verlag, 1984, pp. 225–236.

[55] S. Landau and G. L. Miller, "Solvability by radicals is in polynomial time," in *STOC '83: Proceedings of the fifteenth annual ACM symposium on Theory of computing*. New York, NY, USA: ACM, 1983, pp. 140–151.

[56] S. Laplagne, "An algorithm for the computation of the radical of an ideal," in *ISSAC '06: Proceedings of the 2006 international symposium on Symbolic and algebraic computation*. New York, NY, USA: ACM, 2006, pp. 191–195.

[57] C. E. Lemke and J. T. Howson Jr., "Equilibrium points of bimatrix games," *Journal of the Society for Industrial and Applied Mathematics*, vol. 12, no. 2, pp. 413–423, June 1964.

[58] A. Lenstra, H. Lenstra, and L. Lovsz, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, pp. 515 – 534, 1982.

[59] J. H. W. Lenstra, "Algorithms in algebraic number theory," *Bulletin of the American Mathematical Society*, vol. 26, no. 2, pp. 211–244, April 1992.

[60] R. J. Lipton and E. Markakis, "Nash equilibria via polynomial equations," in *Proc. of the 6th Latin American Symposium on Theoretical Informatics*, Buenos Aires, Argentina, 2004, pp. 413–422.

[61] E. M. Luks, "Permutation groups and polynomial-time computation," in *Groups and Computation II, DIMACS series in Discrete Mathematics and Theoretical Computer Science*, vol. 11, 1993, pp. 139–175.

[62] R. D. McKelvey and A. McLennan, *Handbook of Computational Economics*. Elsevier, 1996, ch. Computation of Equilibria in Finite Games, pp. 87–142.

[63] ——, "The maximal number of regular totally mixed Nash equilibria," *Journal of Economic Theory*, vol. 72, pp. 411–425, 1997.

[64] A. Mclennan, "The expected number of Nash equilibria of a normal form game," *Econometrica*, vol. 73, no. 11, pp. 141–174, 2005.

[65] C. Millham, "Constructing bimatrix games with special properties," *Naval Research Logistics Quarterly*, vol. 19, no. 4, pp. 709–714, 1973.

[66] O. Morgenstern and J. von Neumann, *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[67] R. B. Myerson, "Nash equilibrium and the history of economic theory," *Journal of Economic Literature*, vol. 37, no. 3, pp. 1067–1082, September 1999.

[68] J. Nash, "Non-cooperative games," *The Annals of Mathematics, Second Series, Issue 2*, vol. 54, pp. 286–295, 1951.

[69] J. F. Nash, "Equilibrium points in n-person games," in *Proceedings of the National Academy of Sciences of the United States of America*, vol. 36, no. 1, January 1950, pp. 48–49.

[70] R. Nau, S. G. Canovas, and P. Hansen, "On the geometry of Nash equilibria and correlated equilibria," *International Journal of Game Theory*, vol. 32, pp. 443–453, 2003.

[71] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, Eds., *Algorithmic Game Theory*. New York, NY, USA: Cambridge University Press, 2007.

[72] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. MIT Press, 1999.

[73] C. H. Papadimitriou and T. Roughgarden, "Computing equilibria in multiplayer games," in *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2005, pp. 82–91.

[74] A. Papantonopoulou, *Algebra: Pure and Applied*. Prentice Hall, 2001.

[75] X. Qin, Y. Feng, J. Chen, and J. Zhang, "A Complete Algorithm to Find Exact Minimal Polynomial by Approximations," *CoRR*, vol. abs/1001.0649, 2010.

[76] J. Renegar, "On the efficiency of Newton's method in approximating all zeros of a system of complex polynomials," *Mathematics of Operations Research*, vol. 12, no. 1, pp. 121–148, February 1987.

[77] T. Roughgarden, "Computing equilibria: a computational complexity perspective," *Economic Theory*, vol. 42, no. 1, pp. 193–236, January 2010.

[78] T. Roughgarden and E. Tardos, "How bad is selfish routing?" *Journal of ACM*, vol. 49, no. 2, pp. 236–259, 2002.

[79] R. Segal and R. L. Ward, "Weight distributions of some irreducible cyclic codes," *Mathematics of Computation*, vol. 46, no. 173, pp. 341–354, January 1986.

[80] L. S. Shapley and J. F. Nash, "A simple three-person poker game," *Contributions to the Theory of Games: Annals of Mathematics Study*, vol. 1, no. 24, pp. 105–116, 1950.

[81] S. Smale, "The fundamental theorem of algebra and complexity theory," *Bulletines of American Mathematical Society*, vol. 4, pp. 1–36, 1981.

[82] ——, "Complexity theory and numerical analysis," *Acta Numerica*, vol. 6, pp. 523–551, 1997.

[83] B. Sturmfels, *Solving Systems of Polynomial Equations*, ser. CBMS Regional Conference Series in Mathematics. Providence, Rhode Island: American Mathematical Society, 2002, vol. 97.

[84] A. Vetta, "Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions," in *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002, pp. 416–425.

[85] B. von Stengel, *Computing Equilibria for Two-Person Games*, ser. Handbook of Game Theory with Economic Applications. Amsterdam: Elsevier, 2002, vol. 3, ch. 45, pp. 1723–1759.

[86] J. Zhang and Q. Zhang, "Stackelberg game for utility-based cooperative cognitive radio networks," in *MobiHoc '09: Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*. New York, NY, USA: ACM, 2009, pp. 23–32.