Federal University of the State of Rio de Janeiro
Center for Exact Sciences and Technology
School of Applied Informatics

# An Introduction to Game Theory under Modern Computer Science Lens

Victor Hugo Souza Wirz

**Advisor**
Pedro Nuno de Souza Moura

Rio de Janeiro, RJ – Brasil
February, 2023

Catalogação informatizada pelo(a) autor(a)


W799 Wirz, Victor Hugo Souza
An Introduction to Game Theory under Modern Computer Science Lens / Victor Hugo Souza Wirz. -- Rio de Janeiro, 2023.
65 p.

Orientador: Pedro Nuno de Souza Moura.
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal do Estado do Rio de Janeiro, Graduação em Sistemas de Informação, 2023.

1. Algoritmos. 2. Complexidade Computacional. 3. Teoria dos Jogos. I. Moura, Pedro Nuno de Souza, orient. II. Título.

An Introduction to Game Theory under Modern Computer Science Lens

Victor Hugo Souza Wirz

Undergraduate thesis and graduation project presented at the School of Applied Informatics at Federal University of the State of Rio de Janeiro (UNIRIO) in partial fulfillment of the requirements to obtain the title of Bachelor in Information Systems.

Approved by:

---

Pedro Nuno de Souza Moura, D.Sc. – UNIRIO

---

Adriana Pimenta de Figueiredo, D.Sc. – UNIRIO

---

Andréa Soares Bonifácio, D.Sc. – UNIRIO

Rio de Janeiro, RJ – Brasil

February, 2023

# Acknowledgments

This work is the result of many guidances and both personal and professional influences over the course of my undergraduation. I would like to first and foremost thank my colleagues Rafael Mota, Henrique Rodrigues, Eduardo Barboza, Nathan Aguiar and Thiago Parracho for the moments of joy when university obligations could give us any slack time; as well as my personal friends Vitor Ferreira, for all those afternoons dedicated for recreational mathematics; and Matheus Knup, for always offering one helping hand, sometimes two, back when I didn't even have a personal computer to commit with a computer science course.

Many special thanks go to my colleagues and tutors from IBM Research: Emilio V. Brazil, Rodrigo Ferreira, Priscilla Avegliano, Carlos Raoni and Debora Lima. It lacks words on my lexis to express how transformative and contributing your influences are in the shaping of my professional profile and how easy you made for me to narrow my range of interests when deciding to continue to study and research in the prospect of a graduate program.

I am profoundly grateful to Professor Adriana Pimenta from the Department of Mathematics for the mentorship in Combinatorics as well as on my scientific initiation with the Institute for Pure and Applied Mathematics – IMPA. I also want to thank all of the Department of Applied Informatics from UNIRIO who worked there between 2017 and 2022, from the professors who lectured the most elementary disciplines to those who helped me in more advanced material.

Utmost thanks go to Professor Pedro Moura who supervised this work and was there since my second semester providing me guidance in any subject remotely related to Algorithms and Combinatorial Optimization. I deeply appreciate all the mentorship, and if any day I decide to become a professor, I want to be as helpful to my students as you were to me.

Lastly, but not least, I want to thank my mom, Ericka Coelho, for always being pride of my deeds no matter how weird any idea thought out loud from my mind could strike to one's ears; and finally, to my partner in love Caroline Tavares who now wears such a burden.

*Game Theory is to the social sciences what Calculus is to the physical ones.*

—Pedro Domingos

**Abstract**

Advances in Computer Science for applications of interest in Economics have been growing large, motivated by the ubiquity of the Internet and its role as a public open space used for markets. The field has been driven towards discussions in Microeconomics under game-theoretical frameworks, which aim to model both conflicting and coalitional behavior of interacting agents. In this context, the problem of finding equilibrium points in several types of games has received considerable attention, motivating new theories within Computer Science. This work intends to introduce and formalize the main concepts developed in recent decades to students in Computing-related sciences at the undergraduate and initial graduate level.

**Keywords:** Algorithms, Computational Complexity, Game Theory.

**Resumo**

Avanços em Computação para aplicações de interesse em Economia têm se intensificado, motivados pela ubiquidade da Internet e seu papel como espaço público aberto usado para transações. A área tem se debruçado sobre discussões em Microeconomia sob domínios de Teoria dos Jogos, que busca modelar ambos comportamentos cooperativos e competitivos de agentes em interação. Neste contexto, o problema de encontrar pontos de equilíbrio em diversos tipos de jogos tem recebido atenção substancial, motivando novas teorias em Ciência da Computação. Este trabalho tem por intuito introduzir e formalizar os conceitos basilares desenvolvidos em décadas recentes para estudantes de ciências relacionadas à Computação ao nível de graduação e iniciantes de pós-graduação.

**Palavras-chave:** Algoritmos, Complexidade Computacional, Teoria dos Jogos.

# Contents

**1 Introduction** **1**

1.1 Motivation . . . 1

1.1.1 Why study Game Theory? . . . 1

1.1.2 The Role of Computing . . . 2

1.2 Objective . . . 3

1.3 Method . . . 3

1.4 Text Organization . . . 4

**2 Fundamental Concepts** **5**

2.1 Games . . . 5

2.1.1 What are games? . . . 5

2.1.2 Games on Normal Form . . . 5

2.2 Linear Programming . . . 6

2.2.1 Duality Theory . . . 6

2.3 Zero-sum games and Nash equilibrium . . . 12

**3 Taxonomy of Problems** **18**

3.1 Total Function Problems . . . 19

3.2 Polynomial Local Search . . . 20

3.3 Congestion Games . . . 23

3.4 $\mathcal{PLS}$-completeness . . . 27

3.5 When is PNE tractable? . . . 29

**4 $\mathcal{PPAD}$ and Nash Equilibria** **31**

4.1 From Combinatorics... . . . 31

4.2 To topological statements . . . 35

4.3 General games and Nash equilibrium . . . . . . . . . . . . . . . . . . . . . . . 37

**5 Final Considerations and Conclusions** **41**

5.1 Final Considerations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41

5.2 Future Works . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42

**A Lagrange Multipliers** **43**

**B Linear Algebra** **45**

**C Asymptotic Notation** **46**

**D Definitions from Topology and Real Analysis** **47**

**E Graph Theory** **49**

# List of Figures


1 Abstract polynomial-time reduction . . . . . . . . . . . . . . . . . . . . . . 19

2 Local optimum of a maximal cut instance. . . . . . . . . . . . . . . . . . . . 22

3 Atomic routing game. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

4 Min-cost flow for atomic routing game. . . . . . . . . . . . . . . . . . . . . 30

5 Legally colored simplicial division of a one-dimensional simplex. . . . . . . 31

6 Legally colored simplicial division of a two-dimensional simplex. . . . . . . 32

7 Traversal after introduction of an artificial tri-chromatic triangle. . . . . . . 33

8 Exponentially large graph for END-OF-LINE with vertex set $\{0,1\}^4$. . . . . . 35

9 Two-dimensional probability simplex. . . . . . . . . . . . . . . . . . . . . . 36

10 Two-dimensional probability simplex with colored vertices. . . . . . . . . . 37

11 Compact set of interest for the Penalty game. . . . . . . . . . . . . . . . . . 39

# List of Algorithms


1 Local search for MAX-CUT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21

2 Abstract local search . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

3 Best-response dynamics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

# List of Theorems


Theorem 2.1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

Theorem 2.2 (LP Weak Duality) . . . . . . . . . . . . . . . . . . . . . . . . 8

Corollary 2.2.1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

Corollary 2.2.2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

Theorem 2.3 (Farkas's Lemma, 1902) . . . . . . . . . . . . . . . . . . . . . 9

Corollary 2.3.1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

Theorem 2.4 (Variant of Farkas's Lemma, 1902) . . . . . . . . . . . . . . . . 9

Theorem 2.5 (LP Strong Duality) . . . . . . . . . . . . . . . . . . . . . . . 11

Theorem 2.6 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

Theorem 2.7 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

Proposition 2.3.1 (NASH) . . . . . . . . . . . . . . . . . . . . . . . . . . 16

Theorem 3.1 (Cook-Levin) . . . . . . . . . . . . . . . . . . . . . . . . . . 19

Theorem 3.2 (Rosenthal, 1973) . . . . . . . . . . . . . . . . . . . . . . . . 25

Proposition 3.3.1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

Theorem 3.3 (Schäffer and Yannakakis, 1991) . . . . . . . . . . . . . . . . . 27

Theorem 3.4 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28

Theorem 4.1 (Sperner's Lemma) . . . . . . . . . . . . . . . . . . . . . . . . 31

Lemma 4.2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34

Theorem 4.3 (Brouwer's fixed point) . . . . . . . . . . . . . . . . . . . . . 35

Theorem 4.4 (Nash's Theorem) . . . . . . . . . . . . . . . . . . . . . . . . 38

# 1 Introduction

*Algorithmic Game Theory*, alternatively called *Economics and Computation*, is the study of algorithm design for problems arising in game-theoretical models. According to (Roughgarden, 2016), the interplay between Economics and Computing emerged due to many problems of interest in the modern Computer Science, from resource allocation to online advertising, "fundamentally involving interaction between multiple self-interested parties."

Whilst this area of knowledge has given many fruitful results in the last 15 years, most of the literature available for newcomers is reserved for experienced theoretical computer scientists and mathematicians. This work aims to provide a comprehensive yet-not-exhaustive introduction to students at the undergraduate level whose curiosity might lead them to study such topics.

## 1.1 Motivation

### 1.1.1 Why study Game Theory?

Game Theory can be thought of as an outgrow of Microeconomics, a broader area of Economics concerned with individual markets and industries. Microeconomics deals with the analysis of how economic activity is organized under scarce resources to be allocated among competing uses (Mankiw, 2016). Game Theory can be classified then as a method: the use of mathematical tools in order to model both competitive and coalitional behavior of interacting agents.

Such tools delve deep into the apparatus of Operations Research and serve as guide to business as both policy- and decision-makers:

> Operations research and market research, along with motivational research, are considered crucial and their results assist, in more than one way, in taking business decisions. [...] Research with regard to demand and market factors has great utility in business. Given knowledge of future demand, it is generally not difficult for a firm, or for an industry to adjust its supply schedule within the limits of its projected capacity. Market analysis has become an integral tool of business policy these days. Business budgeting, which ultimately results in a

> projected profit and loss account, is based mainly on sales estimates which in turn depends on business research. Once sales forecasting is done, efficient production and investment programmes can be set up around which are grouped the purchasing and financing plans. Research, thus, replaces intuitive business decisions by more logical and scientific decisions. (Kothari, 2004, page 6)

as well as governments:

> Government has also to chalk out programmes for dealing with all facets of the country's existence and most of these will be related directly or indirectly to economic conditions. The plight of cultivators, the problems of big and small business and industry, working conditions, trade union activities, the problems of distribution, even the size and nature of defence services are matters requiring research. Thus, research is considered necessary with regard to the allocation of nation's resources. (Kothari, 2004, page 6)

### 1.1.2 The Role of Computing

Discrete optimization started to get traction in 1950 when Linear and Integer Programming were formulated and Operations Research as a whole got intensive attention (Schrijver, 2005). Even then, according to (Lawler, 2011), up until the 1970s, combinatorial problems were still thought of to be trivial, "devoid of mathematical content," by pure mathematicians, who would often suggest to enumerate all possible solutions to a given problem and choose the best one according to a selected criteria. Naïve approaches to solve problems of discrete nature, however, may take unreasonable amount of time:

> This line of reasoning is hardly satisfying to one who is actually confronted with the necessity of finding an optimal solution to one of these problems. A naïve, brute force approach simply will not work. Suppose that a computer can be programmed to examine feasible solutions at the rate of one each nanosecond, i.e., one billion solutions per second. Then if there are $n!$ feasible solutions, the computer will complete its task, for $n = 20$ in about 800 years, for $n = 21$ in about 16,800 years, and so on. Clearly, the running time of such a computation is effectively infinite. (Lawler, 2011, page 4)

Likewise for Game Theory, both economists and mathematicians have mainly taken approaches with emphasis in exact solutions and characterizations, with no regards for computational issues. Computer Science has taken the role of investigating procedures – algorithms – capable of solving such problems in an acceptable time, finding approximate solutions when finding exact ones are unrealistic, as well as the limits of the computation involved.

## 1.2 Objective

A textbook is a compilation of content with the purpose to teach a particular subject (OUP, 2018). The objective of this work is to provide a textbook on the study of optimization and algorithms with the purpose to introduce the reader on the discipline of Algorithmic Game Theory.

As this work intends to be an introduction for advanced undergraduate and initial graduate students, at the end of this document the student should be able to understand how the Internet motivates the study of Algorithmic Game Theory; to identify problems of combinatorial nature in Computer Science that fit into game-theoretical frameworks; as well as to identify some properties of combinatorial problems that make $\mathcal{NP}$ theory inadequate. Although not strictly required, the student familiar with Linear Programming or Lagrangean Methods should find it easier to absorb some of the concepts in this monograph.

## 1.3 Method

The method taken in the formulation of this monograph is based on an extensive revision and content analysis regarding topics approaching both the field of Algorithms (Roughgarden, 2016; Bertsimas and Tsitsiklis, 1997; Nisan et al., 2007; Papadimitriou, 2001, 1994) and Game Theory (Leyton-Brown and Shoham, 2008) and Microeconomics (Mankiw, 2016), with occasional related areas filling in the gaps required for further proof of concepts for both subjects, such as Optimization and Topology. We begin by presenting fundamental concepts on both fields and weave them together on the second half of this document under modern Computer Science lens.

## 1.4 Text Organization

The text is organized as follows:

- Chapter 2 first presents the concepts of games, what is a game in normal form, utility function and payoffs, Linear Optimization, equivalence between *zero-sum* games and Duality Theory and the case for general games.
- Chapter 3 is concerned with introducing the concept of taxonomy of problems, describing special sets in Theory of Computation that seek to understand problems for which a solution is guaranteed to exist, the reasoning behind their existence, as well as the concept of completeness and how it relates to Game Theory problems.
- Chapter 4 carries on the notion of totality developed earlier to provide the reasoning of existence of equilibrium for general games and its implications for Computer Science as whole.
- Chapter 5 contains the closing thoughts and final considerations of the work, as well as indications to where future works could improve upon this monograph.

# 2 Fundamental Concepts

This section intends to briefly introduce readers to fundamental concepts needed to understand the problems approached in this work. Readers familiar with the disciplines presented here may skip without loss.

## 2.1 Games

### 2.1.1 What are games?

Games, as in Economics, are situations where agents (players) must interact in some way and each agent has a valuation on the arrangement of the strategic decisions taken by the all players. An example is the traditional ROCK-PAPER-SCISSORS between two players:

| | ROCK | PAPER | SCISSORS |
|---|---|---|---|
| ROCK | $(0, 0)$ | $(-1, 1)$ | $(1, -1)$ |
| PAPER | $(1, -1)$ | $(0, 0)$ | $(-1, 1)$ |
| SCISSORS | $(-1, 1)$ | $(1, -1)$ | $(0, 0)$ |

(2.1)

Bimatrix (2.1) represents a ROCK-PAPER-SCISSORS, each tuple represents the payoff of the row and column players, respectively.

### 2.1.2 Games on Normal Form

A finite, $n$-person game, is defined as tuple $(N, S, u)$:

i. $N$ the set of $n$ players;

ii. $S = S_1 \times \cdots \times S_n$ the cartesian product between available *strategies*, also called *actions*, in $S_i$ for each player $i$;

iii. $u = (u_1, \ldots, u_n)$ the utility functions for each player that maps an action profile $\mathbf{s} \in S$ to a real number, or, more succinctly $u_i : S \mapsto \mathbb{R}$.

The ROCK-PAPER-SCISSORS indicated in the preceding subsection fits this description as a two-player game, where each player has three available strategies (rows and columns of the tableau) and the tuples $(u_1, u_2)$ represent the valuation for each action profile.

Whenever speaking about games, the assumption of rationality is taken, that is: each player aims to maximize its utility function. Beware that this not only means the players are motivated to strive to maximize their utility, it implies that every player always maximizes his utility, thus being able to perfectly calculate the probabilistic result of every action.

## 2.2 Linear Programming

Linear Programming (LP), also known as Linear Optimization, is a class of optimization problems where both the objective function and constraints are linear functions. More generally,

$$\begin{aligned} \min_{x} \quad & c'x \\ \text{s.t.} \quad & x \in P \end{aligned}$$

where $P$ a is polyhedron of the form $P = \{x \in \mathbb{R}^n \mid Ax \lesseqqgtr b\}$. Inequalities in the polyhedron definition can be turned into equalities through the introduction of slack and surplus variables (see Bertsimas and Tsitsiklis, 1997). When $P$ is in the form $P = \{x \in \mathbb{R}^n \mid Ax = b, x \geqslant 0\}$, we say it is in the *standard form*, otherwise it is in the *canonical form*.

Here we give a brief overview of important aspects of the Linear Programming field and its contributions to the Theory of Computation.

### 2.2.1 Duality Theory

Duality Theory is a tool that enable us to analyze the relationship between different instances of optimization problems. Its applicability can be easily generalized for optimization beyond Linear Programming. The explanation here is due to (Bertsimas and Tsitsiklis, 1997), though with a different example, and we refer to it for deeper intuition behind the geometry and economic interpretation of the dual. We start with the problem

$$\begin{aligned} \min_{x} \quad & c'x \\ \text{s.t.} \quad & Ax \geqslant b, \end{aligned}$$

which we call the *primal*, with an optimal solution $x^*$ assumed to exist. By introducing the Lagrange multipliers $p$ with the same dimension as $b$, we can remove the constraints and put them in the objective function.

$$\min_{x} \quad c'x + p'(b - Ax) \tag{L}$$

Under optimal cost, the above problem should have cost no greater than $c'x^*$:

$$\min_x \Big[ c'x + p'(b - Ax) \Big] \leqslant c'x^* + p'(b - Ax^*) = c'x^*$$

Define $g(p)$ to be a function of the lagrangean with respect to $p$ and optimize it to find the infimum of the relation above, we have:

$$\begin{aligned} \max_p g(p) &= \max_p \min_x \Big[ c'x + p'(b - Ax) \Big] \\ &= \max_p \Big[ p'b + \min_x (c' - p'A)x \Big]. \end{aligned}$$

By the second equality, we notice the quantity $p'b$ sets the infimum for the lagrangean for choices of $x$ and, consequently, the primal. Notice, too, since $x$ has no restriction on sign, the only way to ensure the lagrangean amounts to a finite quantity is by restricting values of $p$ such that $p'A = c$. By the first equality, we have $b - Ax$ which we know to be a nonpositive quantity, by construction of the lagrangean, multiplying $p$. Allowing $p$ to take on negative values would imply $\max_p g(p) = \infty$ for any $b - Ax < \mathbf{0}$, so we restrict our choices to $p \geqslant \mathbf{0}$.

The preceding discussion sets down the constraints for our *dual*:

$$
\begin{aligned}
\max_{p} \quad & p'b \\
\text{s.t.} \quad & p'A = c, \\
& \quad p \geqslant \mathbf{0}
\end{aligned}
$$

Repeated construction of the procedure for the dual above leads to the same lagrangean stated in $L$, which proves

**Theorem 2.1.** The dual of the dual is the primal.

One can see that the cost vector $c$ for the primal relates to the dual as constraints and, conversely, the vector $b$ of constraints for the primal relates to the dual as its cost vector. Sign constraints on each of the decision variables for the primal produce inequality and equality constraints on the dual and, conversely, inequality or equality constraints on the dual set up sign constraints in the variables of the primal.

**Theorem 2.2.** (LP Weak Duality) $c'x \geqslant p'b$

*Proof.* Since, by construction, the sign of $c - p'A$ is the same as that of $x$ (or equals 0) and $Ax - b$ is the same as of $p'$ (or, again, equals 0), then both $(c - p'A)x$ and $p'(b - Ax)$ are nonnegative and so is their sum

$$
\begin{aligned}
(c' - p'A)x + p'(Ax - b) &\geqslant 0 \\
c'x - p'Ax + p'Ax - p'b &\geqslant 0 \\
c'x &\geqslant p'b \qquad (2.2)
\end{aligned}
$$

which proves our claim. ∎

The above is a result known as *weak duality*. It is straightforward from inequality (2.2) that:

**Corollary 2.2.1.** If there exists primal-feasible $x$ such that $c'x = p'b$ for any dual-feasible $p$, then such $x$ is optimal for the primal, and so is $p$ for the associated dual.

**Corollary 2.2.2.** If the primal is unbounded, the dual is infeasible.

**Theorem 2.3.** (Farkas's Lemma, 1902) Let $A$ be a matrix of dimensions $m \times n$ and let $b$ be a vector in $\mathbb{R}^m$. The following proposition is true:

$$\nexists\, x \geqslant \mathbf{0} : Ax = b \iff \exists\, p : p'A \geqslant \mathbf{0},\ p'b < 0$$

*Proof.* $(\Longleftarrow)$ Suppose $x \geqslant \mathbf{0} : Ax = b$. If $p'A \geqslant \mathbf{0}$, then, multiplying both sides by $x$ yields $p'Ax \geqslant 0$. Since both $p'A$ and $x$ are assumed to be $\geqslant \mathbf{0}$, $p'Ax = p'b < 0$ can't hold.

$(\Longrightarrow)$ Suppose the set $Ax = b, x \geqslant \mathbf{0}$ to be empty. We define the cone

$$Q = \{y \mid Ax = y, x \geqslant \mathbf{0}\}$$

into which $b$ is not contained. By application of the hyperplane separating theorem, we can define $p$ such that $p'b < p'y$ for all $y \in Q$.

We first note that since $\mathbf{0} \in Q$, then $p'b < 0$. Denoting $A_i$ as the $i$th column of $A$, $\lambda A_i \in Q$, $\lambda > 0$ – to see this consider[1] $x = \lambda e_i$. So,

$$\begin{aligned} p'b &< \lambda p'A_i \\ \frac{p'b}{\lambda} &< p'A_i \\ \lim_{\lambda\to\infty} \frac{p'b}{\lambda} &< p'A_i \\ 0^- &< p'A_i, \end{aligned}$$

where $0^-$ means that the limit approaches zero by the left.

Since this is true for all $i$, we can conclude $p'A \geqslant \mathbf{0}'$. ∎

**Corollary 2.3.1.** Let $p'b \geqslant 0$ and $p'A \geqslant \mathbf{0}$, $b$ can be expressed as a conic combination of the columns of $A$.

**Theorem 2.4.** (Variant of Farkas's Lemma, 1902) Let $A$ be a matrix of dimensions $m \times n$ and let $b$ be a vector in $\mathbb{R}^m$. It is true that:

$$\nexists\, x : Ax \geqslant b \iff \exists\, p \geqslant \mathbf{0} : p'A = \mathbf{0}',\ p'b > 0$$

1. We use the notation $e_i$ to denote a vector of any dimension with the $i$-th entry set to 1 and the remaining set to 0.

*Proof.* ($\Longleftarrow$) Suppose $Ax \geqslant b$. If $p'A = \mathbf{0}'$ and $p \geqslant \mathbf{0}$, then, multiplying both sides by $p'$ yields $p'Ax \geqslant p'b$. Since $p'A = \mathbf{0}'$, $p'b > 0$ can't hold.

($\Longrightarrow$) Suppose the set $Ax \geqslant b$ to be empty. We define the polyhedron

$$Q = \{y \mid Ax = y\}$$

for which $b$ is not defined. By application of the hyperplane separating theorem, we define $p$ and $\epsilon$ such that

$$p'b > \epsilon \text{ and } p'y \leqslant \epsilon.$$

We first note that since $\mathbf{0} \in Q$, then $p'b > \epsilon \geqslant 0$. Denoting $A_i$ as the $i$th column of $A$, for $x = \lambda e_i$, $\lambda > 0$. We have,

$$\begin{aligned}
\lambda p'A_i &\leqslant \epsilon \\
p'A_i &\leqslant \frac{\epsilon}{\lambda} \\
p'A_i &\leqslant 0 \text{ (as } \lambda \to +\infty)
\end{aligned}$$

Analogously, for $x = \lambda e_i$, $\lambda < 0$.

$$\begin{aligned}
\lambda p'A_i &\leqslant \epsilon \\
p'A_i &\geqslant \frac{\epsilon}{\lambda} \\
p'A_i &\geqslant 0 \text{ (as } \lambda \to -\infty)
\end{aligned}$$

Since this is true for all columns of $A$, we conclude then $p'A = 0$.

Finally, for any vector $y \in Q$ this means $\exists\, i : y_i < b_i$, so we can add to $y$ a vector $s \geqslant \mathbf{0}$ such that $y_i + s_i$ lies closer to the halfspace[2] $a_i'x \geqslant b_i$ potentially lying on the boundary $p'v = \epsilon$ or on the other side of the hyperplane. Thus, for vector $s$ with sufficient

2. by $a_i$ we mean the $i$th row vector of $A$

large entries,

$$
\begin{aligned}
p'(y+s) &\geqslant \epsilon \\
p'y + p's &\geqslant \epsilon \\
p's &\geqslant \epsilon - p'y \\
p's &\geqslant 0
\end{aligned}
$$

As $s \geqslant \mathbf{0}$, so must be $p$ for $p's \geqslant 0$ to be satisfied. ■

Farka's lemma tells us feasibility in one system is a certificate of infeasibility in another, and it is useful to prove the following theorem without relying on exploitation of termination rules of iterative algorithms that solve LP.

**Theorem 2.5.** (LP Strong Duality) Whenever an optimum $x^*$ exists in the primal, there exists a $p^*$ optimal in the dual. Moreover, $c'x^* = (p^*)'b$.

*Proof.* Let the following pair of primal-dual be our problems under consideration.

$$
\begin{aligned}
\min_{x} \quad & c'x \\
\text{s.t.} \quad & Ax \geqslant b
\end{aligned}
\qquad\qquad
\begin{aligned}
\max_{p} \quad & p'b \\
\text{s.t.} \quad & p'A = c, \\
& p \geqslant \mathbf{0}
\end{aligned}
$$

Suppose both primal and dual to have feasible solutions, the dual holding optimal $h$. By weak duality, it's trivial that if we prove the existence of feasible $x^*$ such that $c'x^* \leqslant h$, then $x^*$ is an optimal solution to the primal. By hypothesis we neglect the existence of this solution, then by Farkas's lemma,

$$
\nexists x : \begin{bmatrix} A \\ -c' \end{bmatrix} \begin{bmatrix} x \end{bmatrix} \geqslant \begin{bmatrix} b \\ -h \end{bmatrix} \iff \exists\, p \geqslant \mathbf{0}, \lambda \geqslant 0 : \begin{bmatrix} p \\ \lambda \end{bmatrix}' \begin{bmatrix} A \\ -c' \end{bmatrix} = \mathbf{0}', \begin{bmatrix} p \\ \lambda \end{bmatrix}' \begin{bmatrix} b \\ -h \end{bmatrix} > 0
$$

which results in $p'A = \lambda c'$ and $p'b > \lambda h$.

If we assume $\lambda = 0$, then $p'A = \mathbf{0}'$ and $p'b > 0$, which implies the inexistence of $x$ such that $Ax \geqslant b$, contradicting our assumption of feasibility for the primal. For $\lambda > 0$, for $\bar{p}$ to satisfy $\bar{p}'A = c$ it must be that $\bar{p} = \frac{1}{\lambda}p$, which is feasible by nonnegativity of $p$

and $\lambda$, however, $\frac{1}{\lambda}p'b > h$ which contradicts the assumption that $h$ is the optimal in the dual.

We conclude $\exists\, x \,:\, Ax \geqslant b$ with $c'x \leqslant h$. As $h$ is the optimal for the dual, it follows from weak duality that $c'x = h$. ■

Theorem 2.5 a is remarkable result and it allows us to show that optimizing a linear program is no harder than answering if a polyhedron is nonempty. Let $P := \{x \mid Ax \geqslant b\}$ the polyhedron defined for the primal and $Q := \{p \mid p'A = c, p \geqslant 0\}$ the polyhedron defined for the dual. We need to find a solution for systems of linear inequalities

$$\begin{aligned} c'x &= p'b & \qquad Ax &\geqslant b \\ p'A &= c & \qquad p &\geqslant \mathbf{0} \end{aligned}$$

In other words, if there is an oracle that given the polyhedron $S = \{x \in P, p \in Q \mid c'x = p'b\}$ is capable of correctly finding a point in S or reports that none exists, this oracle solves the linear programs under considerations.

## 2.3 Zero-sum games and Nash equilibrium

In competitive games, there's a special type of game called *constant sum games*, where for any strategy profile the sum of all player's payoffs is equal to some constant. For two-player games, in order to simplify, we fix this constant at 0 and let the payoff of player 2 be the negative of player 1. Such games are called *zero-sum games*.

In order to illustrate this concept, we work on a resolution to Exercise 4.10 proposed in (Bertsimas and Tsitsiklis, 1997), stated as:

> Consider the standard form problem of minimizing $c'x$ subject to $Ax = b$ and $x \geqslant \mathbf{0}$. We define the *Lagrangean* by
>
> $$L(x, p) = c'x - p'(Ax - b).$$
>
> Consider the following "game": player 1 chooses some $x \geqslant 0$, and player 2 chooses some $p$; then, player 1 pays to player 2 the amount $L(x, p)$. Player 1 would like to minimize $L(x, p)$, while player 2 would like to maximize it. A pair $(x^*, p^*)$, with $x^* \geqslant 0$, is called an *equilibrium point* (or a *saddle point*, or

a *Nash equilibrium*) if

$$L(x^*, p) \leqslant L(x^*, p^*) \leqslant L(x, p^*),\ \forall x \geqslant \mathbf{0},\ \forall p.$$

Show that a pair $(x^*, p^*)$ is an equilibrium if and only if $x^*$ and $p^*$ are optimal solutions to the primal under consideration and its dual, respectively.

$$\begin{aligned} \min_{x} \quad & c'x \\ \text{s.t.} \quad & Ax = b, \qquad (\mathcal{P}) \\ & x \geqslant \mathbf{0} \end{aligned} \qquad\qquad \begin{aligned} \max_{p} \quad & p'b \\ \text{s.t.} \quad & p'A \leqslant c, \qquad (\mathcal{D}) \end{aligned}$$

**Theorem 2.6.** The equilibrium pair of strategies $(x^*, p^*)$ is precisely the corresponding solutions to the primal $\mathcal{P}$ and its dual $\mathcal{D}$.

*Proof.* $\big(\implies\big)$ Let $(x^*, p^*)$ be a saddle point. So that:

i. $L(x^*, p) \leqslant L(x^*, p^*)$

ii. $L(x^*, p^*) \leqslant L(x, p^*)$

We need to show $x^*$ and $p^*$ constitute optimal solutions to primal and dual.

For i.,

$$c'x^* + p'(b - Ax^*) \leqslant c'x^* + (p^*)'(b - Ax^*)$$
$$p'(b - Ax^*) \leqslant (p^*)'(b - Ax^*)$$

can only hold with equality by choosing $x^*$ such that $Ax^* = b$.

For ii.,

$$c'x^* + (p^*)'(b - Ax^*) \leqslant c'x + (p^*)'(b - Ax)$$
$$(p^*)'b + (c - (p^*)'A)x^* \leqslant (p^*)'b + (c - (p^*)'A)x$$
$$(c - (p^*)'A)x^* \leqslant (c - (p^*)'A)x$$

holds by choosing $x_i^* = 0 \; \forall i \, : \, a_i' p^* < c_i$ and letting $x_i \to +\infty$ on the right-hand side whilst $c - (p^*)'A \geqslant 0$. We conclude $x^*$ is optimal solution to $\mathcal{P}$.

After establishing $(c - (p^*)'A)x^* = 0$, it follows that

$$(p^*)'b \geqslant p'b + (c - p'A)x^* \geqslant p'b.$$

Thus, $p^*$ is optimal solution to $\mathcal{D}$.

$(\Longleftarrow)$ Let $x^*$ and $p^*$ be optimal solutions to $\mathcal{P}$ and $\mathcal{D}$:

For $L(x^*, p)$ and $L(x^*, p^*)$, by optimality of $x^*$ for the primal, we have that $Ax^* = b$, so the Lagrangeans

$$L(x^*, p) = L(x^*, p^*) = c'x^*.$$

So proposition i. is true.

For $L(x, p^*)$, rearranging the Lagrangean to

$$L(x, p^*) = (c' - p^*A)x + (p^*)'b$$

by optimality of $p^*$ for the dual and $x \geqslant \mathbf{0}$, $(c' - p^*A)x$ amounts to a nonnegative quantity and, by strong duality in Theorem 2.5, $(p^*)'b = c'x^*$. So

$$L(x, p^*) = (c' - p^*A)x + c'x^*$$

and therefore

$$L(x^*, p) \leqslant L(x^*, p^*) \leqslant L(x, p^*).$$

So if player 1 fails to choose $x^*$ whereas player 2 is sucessful in choosing $p^*$, she will pay $(c' - p^*A)x$ more than the minimum that could be paid. Conversely, when player 1 chooses $x^*$, player 2 is indifferent about which strategy to play. In other words: no player can increase its utility by unilaterally modifying their strategy. We call $L(x^*, p^*) = v$ the *value of the game*. ■

The proof of Theorem 2.6 shows the equivalence between zero-sum games and LP duality. In fact, the minimax theorem first proved by John von Neumann in 1928 through means of fixed-point arguments comes out naturally as a corollary of strong duality (Stengel, 2022). In simpler terms, let $A \in \mathbb{R}^{m\times n}$ be the payoff matrix of the first player, such that she needs to assign a probability distribution $x$ over the rows of $A$. Likewise, the second player needs to assign a probability distribution $y$ over the columns of $A$. The expected payoff for the first player is $x'Ay$ and $\max_x \min_y x'Ay$ is the value of the game. Moreover,

$$\max_x \min_y \; x'Ay = \min_y \max_x \; x'Ay = v,$$

to see why this is true – and why the seemingly quadratic nature of optimization in $x'Ay$ is of no value – observe that once the player who goes first decides a strategy, the opponent needs not to randomize, but simply choose the best response deterministically.

**Theorem 2.7.** $\max\limits_x \min\limits_y \; x'Ay = \min\limits_y \max\limits_x \; x'Ay$

*Proof.*

$$\max_x \min_y \; x'Ay = \max_x \left( \min_{j=1}^{n} \; (x'A)_j \right) \tag{2.3}$$

$$= \max_x \left( \min_{j=1}^{n} \; \sum_i^m x_i a_{ij} \right) \tag{2.4}$$

Equation (2.4) can be stated as the following linear program:

$$\begin{aligned} \max_{q,\, x} \quad & q \\ \text{s.t.} \quad & \mathbf{1}q - x'A \leqslant \mathbf{0}, \\ & \mathbf{1}'x = 1, \\ & x \geqslant \mathbf{0}, \\ & x \in \mathbb{R}^m, \\ & q \in \mathbb{R} \end{aligned} \tag{$\mathcal{P}'$}$$

By solving this linear program, player 1 arrives at the strategy vector $x^*$ that maximizes his payoff when player 2 plays optimally. It is straightforward from $\mathcal{P}'$ that $q$ attains maximum equal to $\min\limits_{j=1}^{n} \; \sum_i^m x_i a_{ij}$. Doing the same for $\min_y \max_x \; x'Ay$ results in the dual of $\mathcal{P}'$, which by strong duality proves our claim. ■

Getting back to our example with Rock-Paper-Scissors, the payoff matrix for the row player is

$$A = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix}.$$

Whereas the payoff matrix for the column player is $-A$. Each of them aims to maximize their respective payoffs, the row player faces the problem $\max_x \min_y x'Ay$, and column player, $\max_y \min_x x'(-A)y = -\min_y \max_x x'Ay$.

After solving the linear program

$$\begin{aligned} \max_{q,\,x} \quad & q \\ \text{s.t.} \quad & \qquad x_2 - x_3 \geqslant q, \\ & -x_1 \qquad + x_3 \geqslant q, \\ & x_1 - x_2 \qquad \geqslant q, \\ & x_1 + x_2 + x_3 = 1, \\ & \qquad x_i \geqslant 0 \qquad i = 1, 2, 3 \end{aligned} \tag{Row player}$$

the row player maximizes his payoff with uniform probability distribution over the rows of $A$, i.e., $x_1^* = x_2^* = x_3^* = \frac{1}{3}$. Conversely, the column player solves

$$\begin{aligned} \min_{t,\,y} \quad & t \\ \text{s.t.} \quad & \qquad y_2 - y_3 \leqslant t, \\ & -y_1 \qquad + y_3 \leqslant t, \\ & y_1 - y_2 \qquad \leqslant t, \\ & y_1 + y_2 + y_3 = 1, \\ & \qquad y_i \geqslant 0 \qquad i = 1, 2, 3 \end{aligned} \tag{Column player}$$

and also maximizes his payoff with uniform probability distribution over the columns of $A$.

We leave the discussion for games different than zero-sum to Chapters 3 and 4, but state the following proposition here in advance:

**Proposition 2.3.1.** (NASH) If the game is finite, a Nash Equilibrium always exists.

We also point out that for two-player nonzero-sum games, equilibrium computation is equivalent to quadratic programming (Mangasarian and Stone, 1964), but we leave this discussion out the scope of this monograph.

# 3 Taxonomy of Problems

In Computing, the standard way to classify problems is usually in the sense of $\mathcal{P} \stackrel{?}{=} \mathcal{NP}$ question. A problem is said to be in $\mathcal{P}$ – which stands for POLYNOMIAL – if and only if there is a deterministic algorithm which solves any instance of that problem with time growth bounded by a polynomial function in the length of the input. A problem is said to be in $\mathcal{NP}$ – NONDETERMINISTIC POLYNOMIAL – if the validity of a certificate to a problem instance can be verified in polynomial time in the length of the input.

Of course, $\mathcal{P} \subset \mathcal{NP}$ since given any certificate to a problem in $\mathcal{P}$ we might just compute the solution and compare it to the certificate. The open problem is whether $\mathcal{P}$ is a proper subset of $\mathcal{NP}$ or not.

There exists a sense of "hierarchy" inside $\mathcal{NP}$ called $\mathcal{NP}$-completeness. A problem is said to be $\mathcal{NP}$-complete if and only if:

i. It belongs to $\mathcal{NP}$;

ii. all problems in $\mathcal{NP}$ are reducible to it through a polynomial-time reduction.

If ii. is true and i. is not, the problem is said to be $\mathcal{NP}$-hard.

Completeness results are important in the discussion of Theory of Computation because if one could come up with a poly-time algorithm to solve any $\mathcal{NP}$-complete problem then all those in $\mathcal{NP}$ could be solvable in polynomial time in the length of its input. Up until now, no such algorithm is known – and neither a proof that $\mathcal{P} \neq \mathcal{NP}$.

The canonical problem from which $\mathcal{NP}$ problems arise is the *satisfiability* problem, shortened to SAT. The reduction of any problem in $\mathcal{NP}$ to SAT is done by clever construction of a boolean circuitry that translates possible computations of a *nondeterministic Turing machine* to boolean expressions that are satisfied if and only if the Turing Machine accepts the language that represents the instance of the problem. We state the theorem here without a proof, and the reader is referred to (Sipser, 2012) for detailed explanation.

**Satisfiability.** (SAT) Given a boolean expression in clausal normal form (CNF) composed of $n$ variables, report whether exists a boolean assignment for each variable such that the given expression evaluates to 1 (*true*) or not.

**Theorem 3.1.** (Cook-Levin) SAT is $\mathcal{NP}$-complete.

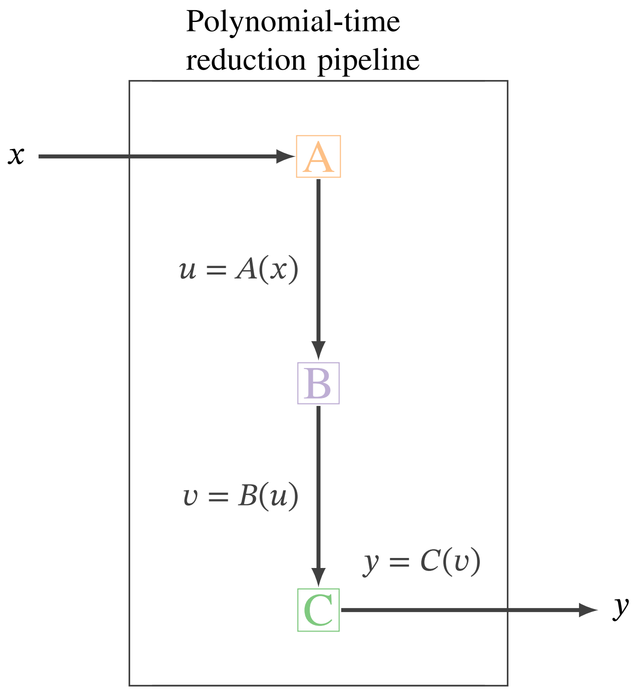


Figure 1: General procedure for reducing one problem to another in polynomial time.

Figure 1 illustrates how reductions work. A problem instance $x \in L$ is fed as input to Algorithm A, which maps $x$ to an instance $u \in L'$. Algorithm B computes certificates for problems in $L'$, and returns $v$ such that $v$ is a certificate for $u$. Algorithm C maps $v$ to a valid certificate $y$ such that $y$ is a solution to $x$. All algorithms are assumed to run in polynomial time and produce outputs bounded by polynomials in the length of their respective inputs.

## 3.1 Total Function Problems

The notion briefly discussed above only applies to decision problems, where we expect an answer YES or NO and much of the difficulty in solving such problems lies on the uncertainty of the existence of a valid solution. It doesn't make much sense to ask whether there is a Nash equilibrium in a game or not, the answer is always positive – and if one could come up with a reduction of SAT to NASH the answer would not translate back to a valid answer for SAT, since a satisfiable assignment of truth values for an expression might not exist. Just as NASH, there is a universe of problems for which a solution is guaranteed to exist. We collectively refer to such problems as TOTAL FUNCTION PROBLEMS.

**Definition.** A *search problem* $L$ is defined by a relation $R_L \subseteq \{0,1\}^* \times \{0,1\}^*$ such that $(x, y) \in R_L$ if and only if $y$ is a solution to $x$.

A search problem $L$ is called *total* if and only if $\forall x\ \exists y : (x, y) \in R_L$.

**Definition.** A problem $L$ is said to be in $\mathcal{FNP}$ – FUNCTION NONDETERMINISTIC POLYNOMIAL – if and only if there exists a polynomial-time algorithm $A_L(\cdot,\cdot)$ and polynomial function $p_L(\cdot)$ such that

i. $\forall x, y\ A(x, y) = 1 \iff (x, y) \in R_L$

ii. $\forall x : \exists y\ : (x, y) \in R_L \implies \exists z : |z| \leq p_L(|x|) \wedge (x, z) \in R_L$

Statement i. asserts a verifier algorithm $A_L$ must be capable of correctly identifying valid certificates for problem instance $x$. Statement ii. asserts for any solution $y$ to a problem instance $x$ there must be a certificate representation $z$ whose length is polynomially bounded by the length of $x$.

**Definition.** (Total $\mathcal{FNP}$) $\mathcal{TFNP} = \{L \in \mathcal{FNP} \mid L \text{ is total}\}$.

Whilst a convenient definition, $\mathcal{TFNP}$ is still a difficult class to study as whole, mainly because the argument behind the existence of a solution for different problems might differ. Due to this, $\mathcal{TFNP}$ is sometimes called a "semantic class." Different arguments providing existence of solutions call for an analog of $\mathcal{NP}$-completeness for other domains of problems. Next, we discuss classes that relate to the concept of equilibrium in Game Theory.

## 3.2 Polynomial Local Search

The prime problem to introduce the class $\mathcal{PLS}$, Polynomial Local Search, is the MAX-CUT. Some definitions are required:

**Definition.** A network $\mathcal{G} = (V, E)$ is a connected graph featuring edges weighted with nonnegative numbers which we refer to as capacities.

**Definition.** A cut is a partition of $V$ into subsets[3] $A$ and $A^c$.

We call the sum of capacities from edges connecting vertices from $A$ to $A^c$ the *cut capacity*. If we visualize the capacity of a cut as the cost of making the graph disconnected, we have the following problem:

> **Maximal Cut Problem.** Given a network $\mathcal{G}$, find a cut such that its capacity is at least the size of any other, i.e., a maximal cut.

MAX-CUT is an $\mathcal{NP}$*-hard* problem, therefore a polynomial-time algorithm in the size of $\mathcal{G}$ that finds the optimal solution inexists unless $\mathcal{P} = \mathcal{NP}$ (Garey and Johnson, 1990). *Local search* is an heuristic utilized to provide feasible solutions for many problems in $\mathcal{NP}$-hard (Papadimitriou and Steiglitz, 1998), including the MAX-CUT.

---

**Algorithm 1:** Local search for MAX-CUT

---

```
Input: G = (V, E)
Output: (A, A^c) cut
begin
    start with an arbitrary cut (A, A^c)
    while there is local movement that improves the objective function do
        take such a movement
    end
    return (A, A^c)
end
```

---

By *local movement* here, we mean: moving to $A^c$ a vertex $v$ currently in $A$ and vice versa. The cost of the solution (when moving a vertex from $A$ to $A^c$ ) increments by an additive factor

$$\sum_{u \in A : (u,v) \in E} c_{u,v} - \sum_{u \in A^c : (u,v) \in E} c_{u,v}. \tag{3.1}$$

If this difference is positive, then it is a valid local movement. For the special case where all edges capacities are the same, let it be 1, the objective function takes values in the set $\{0, 1, \dots, |E|\}$ and updates the feasible solution at hand only when finding values strictly

3. Usually partition sets are required to be nonempty, but allowing so doesn't affect the maximal cut capacity.

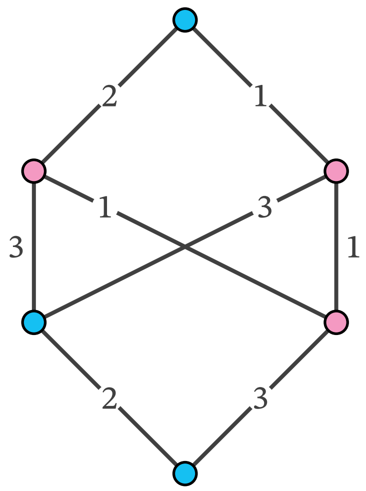

Figure 2: Local optimum of a maximal cut instance: $v \in A$ are colored in blue, $v \in A^c$ colored in pink.

greater for the objective function, the algorithm takes no more than $|E|$ iterations. However, for the general case with nonnegative edges, the algorithm may take exponential time in the number of vertices. Figure 2 illustrates a local maximum for a MAX-CUT instance with partitions being denoted by vertices' colors. One can note that exchanging two vertices between partitions achieve global maximum, but no local movement can increment cut capacity; ergo, Algorithm 1 halts returning the illustrated partition.

If we interpret the steps of computation the algorithm takes over the local search execution as vertices on a graph, we have a directed acyclic graph (*dag*) with each vertex representing a possible cut for $\mathcal{G}$. Directed edges to other vertices represent the local movements available, the vertices representing local optima being *sinks* – vertices with no outgoing edge – and the initial feasible solution being a *source* – vertex with no ingoing edge. Such a graph is sometimes called *transition graph* in the literature.

> **Sink of dag.** Given a source vertex in a directed acyclic graph (*dag*) $\mathcal{G}$, find a vertex with no outgoing edges.

Although the problem might seem trivial by means of usual traversal algorithms, it is not the case here. For each vertex in $\mathcal{G}$ there are two possible states when it comes to making a cut: it either belongs to the partition $A$ or not, making a total of $2^{|V|-1}$ possible cuts. Thus we say the graph defined for SINK-OF-DAG is *exponential in the description of the problem* MAX-CUT and Algorithm 1 may take exponential time in the size of $\mathcal{G}$.

**Algorithm 2:** Abstract local search

**Input:** local search instance
**Output:** a local optimum

**begin**
  start with an arbitrary feasible solution $W$
  **while** there is feasible $W'$ with better cost on the neighboorhood of $W$ **do**
    $W := W'$
  **end**
  **return** $W$
**end**

**Definition.** A problem $L$ is said to be in $\mathcal{PLS}$ if there is a polynomial-time algorithm that reduces $L$ to a SINK-OF-DAG instance.

Since, by definition, a dag must have at least one source (a vertex with no incident edges) and a least one sink (a vertex with no outgoing edges), a solution is guaranteed to exist. Thus, SINK-OF-DAG belongs to $\mathcal{TFNP}$.

Algorithm 2 gives the general procedure for a local search. It is composed of three implicit subroutines: verification of feasibility of a solution; inspection of neighboring solutions[4]; and computation of a solution's cost. All assumed to run in polynomial time in the length of the input.

## 3.3 Congestion Games

Congestion games are a special type of game where each player has its strategy attached to a set $E$ of resources, each resource with an accompanying cost function. The payoff of a player being a function over the subset of resources each player has chosen. So along with the tuple defined for the usual normal-form games, we consider:

i. a set $E$ of congestible resources;

ii. strategy sets $S_i \subseteq 2^E$ for each player $i$;

iii. $f_e$ the number of agents that chooses an element $e \in E$;

iv. the cost function $c_e(\cdot)$ for a congestible resource $e$.

4. We call *neighborhood* the set of solutions that differ from a given solution by a minimal possible extent.

For instance, resources can be edges in a flow network. The following flow network defines the Bimatrix (3.2) for a two-player game. Each player has to decide which route to take from $s$ to $t$.

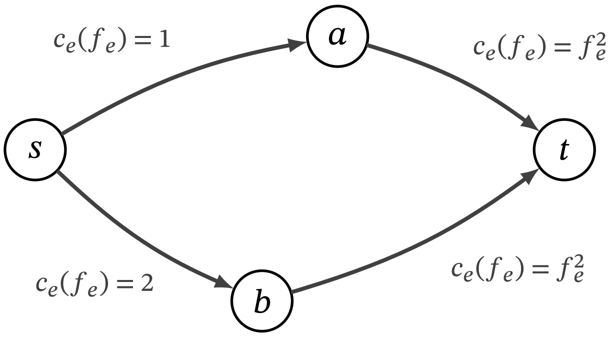


Figure 3: Atomic routing game: routing through $a$ has payoff $1 + f_e^2$ and routing through $b$ has payoff $2 + f_e^2$.

If both route through $a$, then $f_e = 2$ for the edges $(s, a)$ and $(a, t)$, such that the delay for them both is $c_{s,a}(f_{s,a}) + c_{a,t}(f_{a,t}) = 1 + 4 = 5$. Similarly, if both route through $b$, $f_e = 2$ for the edges $(s, b)$ and $(b, t)$, with total delay $c_{s,b}(f_{s,b}) + c_{b,t}(f_{b,t}) = 2 + 4 = 6$.

However, in case they route through different paths, $f_{s,a} = f_{a,t} = 1$ and $f_{s,b} = f_{b,t} = 1$, so one who goes through $a$ gets 2 units of total delay, whereas the one who goes through $b$ gets 3.

The particular case for congestion games with traffic networks is called *routing games* and can be further categorized in *atomic*, in which each player have major contribution to the flow state in the network, and *nonatomic*, in which each player individually have negligible effect on the flow state. Nash equilibrium in such games is also called equilibrium flow (Roughgarden, 2016).

| | $a$ | $b$ |
|---|---|---|
| $a$ | $(5, 5)$ | $(2, 3)$ |
| $b$ | $(3, 2)$ | $(6, 6)$ |

(3.2)

In the atomic routing game depicted by the Bimatrix (3.2), it can be seen the pair of strategies $(a, b)$ and $(b, a)$ constitute pure Nash equilibria, since no player can reduce his cost by unilaterally changing the strategy profile. We introduce notation now to make the concept of Nash equilibrium illustrated in Theorem 2.6 succint.

Let $N$ be the set of players and $S_i$ the strategy set for each player $i \in N$. We denote the strategy player $i$ picks by $s_i \in S_i$ and collectively refer to strategies chosen by other players as $\mathbf{s}_{-i}$, such that the strategy profile is $\mathbf{s} = (s_i, \mathbf{s}_{-i})$.

**Definition.** (Pure Nash Equilibrium) A strategy profile $\mathbf{s} = (s_i, \mathbf{s}_{-i})$ of a cost-minimization game with cost function $C_i$ for each player $i \in N$ is a *pure Nash equilibrium (PNE)* if

$$C_i(\mathbf{s}) \leqslant C_i(s_i', \mathbf{s}_{-i}), \forall i$$

and every unilateral deviation $s_i' \in S_i$.

The definition of PNE for profit-maximization game is analogous, changing the direction of the inequality.

We say that $s_i$ is a *best response* to $\mathbf{s}_{-i}$.

**Theorem 3.2.** (Rosenthal, 1973) Every congestion game has at least one pure Nash equilibrium.

*Proof.* Let $\mathbf{s}$ be an assignment of strategies for each player.

We define the *potential function* (also known as *Rosenthal's function*)

$$\phi(\mathbf{s}) = \sum_{e \in E} \sum_{i=1}^{f_e} c_e(i). \tag{3.3}$$

When a given player $i$ unilaterally deviates from $\mathbf{s}$ by changing its congestible-element set $s_i$ to $s_i'$ we have that

$$\begin{aligned} \phi(s_i', \mathbf{s}_{-i}) - \phi(\mathbf{s}) &= \sum_{e \in s_i' \setminus s_i} c_e(f_e + 1) - \sum_{e \in s_i \setminus s_i'} c_e(f_e) \\ &= \sum_{e \in s_i'} c_e(f_e') - \sum_{e \in s_i} c_e(f_e) \\ &= C_i(s_i') - C_i(s_i). \end{aligned} \tag{3.4}$$

Here $f_e'$ is the number of players picking resource $e$ for strategy profile $(s_i', \mathbf{s}_{-i})$. The change in $\phi$ is the change in player $i$'s cost when deviating. Since there is a finite number of assignments of congestible resources, there is a minimum for $\phi$. ■

One shall note that $\phi(\cdot)$ on its own has no meaningful interpretation, it just happens that the change for $\phi$ under unilateral deviation is equal to the deviator's change in payoff. Any game with the property of being associated with a function of the form of equation (3.4) is called *potential game*, and its PNE can be found by minimizing Rosenthal's function (3.3).

---

**Algorithm 3:** Best-response dynamics

---

**Input:** a game of any form
**Output:** a PNE

**begin**
  start with an arbitrary strategy profile $\mathbf{s}$
  **while** the outcome of $\mathbf{s}$ is not a PNE **do**
    pick an arbitrary agent $i$ with an arbitrary beneficial deviation $s_i'$
    $\mathbf{s} := (s_i', \mathbf{s}_{-i})$
  **end**
  **return s**
**end**

---

Finally, we can show that congestion games belong to $\mathcal{PLS}$ by computing a local minimum through means of *best-response dynamics* – Algorithm 3, the analogous for local search in Game Theory.

**Proposition 3.3.1.** In potential games, regardless of initial strategy profile, best-response dynamics always converge to a PNE.

*Proof.* Since the loop iterates strictly when there is a benefit for the deviator, the change in the potential function $\phi$ as per equation (3.4) is always negative. By finiteness of the game, there is a minimum and Algorithm 3 halts. ■

For general games, one cannot expect Algorithm 3 to terminate in a game with no PNE – e.g., Rock-Paper-Scissors, but even if there is a PNE in the game, best-response dynamics might still get caught in a cycle. In the min-cost game of Bimatrix (3.5), the PNE is the strategy profile (S, S), but if Algorithm 3 starts from (R, R) it loops indefinitely through (R, R), (R, P), (P, P) and (P, R). It can never directly go to (S, S) because it is never a unilateral deviation, nor go into (S, R), (S, P), (R, S) or (P, S) because no player can get a negative change in payoff.

| | R | P | S |
|---|---|---|---|
| R | $(-1,1)$ | $(1,-1)$ | $(1,1)$ |
| P | $(1,-1)$ | $(-1,1)$ | $(1,1)$ |
| S | $(1,1)$ | $(1,1)$ | $(-1,-1)$ |

(3.5)

Regarding the number of iterations Algorithm 3 might take, we must notice that in an $n$-player game, assuming each player to have a $d$ number of strategies, our $n$-matrix would feature $d^n$ entries corresponding to strategy profiles, and each entry have $n$ numbers to describe the payoff of each player, totaling $nd^n$ numbers. Each iteration requires checking whether there's a player with a better response to the current strategy profile $\mathbf{s}$, bounding Algorithm 3 to $\mathcal{O}(nd^{n-1})$ evaluations of whether or not $\mathbf{s}$ is a PNE, with $\mathcal{O}(d^{n-1})$ local movements in the induced dag of the search problem – another case where the transition graph might be exponentially large in the description of the problem just like Max-Cut.

## 3.4 $\mathcal{PLS}$-completeness

We started the discussion stating that Max-Cut is the canonical problem from which $\mathcal{PLS}$ arises, but defined $\mathcal{PLS}$ in terms of Sink-of-Dag. Thereby, Sink-of-Dag is our *complete* problem for $\mathcal{PLS}$ by definition. Formally, to say Max-Cut is also $\mathcal{PLS}$-complete we need to provide a polynomial-time reduction from Sink-of-Dag to Max-Cut.

Proving so requires proving another two problems, Circuit-Flip and Pos-Not-All-Equal-3Sat[5], to be $\mathcal{PLS}$-complete and then providing a reduction from Pos-Not-All-Equal-3Sat to Max-Cut. This was first proved by (Schäffer and Yannakakis, 1991) on a seminal paper showing several $\mathcal{PLS}$ problems to be hard to solve. We do not show the procedure here as it departs too much from our discussion on Game Theory. Readers interested in $\mathcal{PLS}$-reductions can also see (Borzechowski, 2016) for an extensive treatment.

**Theorem 3.3.** (Schäffer and Yannakakis, 1991) All problems in $\mathcal{PLS}$ are polynomial-time reducible to Max-Cut.

5. Positive-weighted not-all-equal 3-Satisfiability.

Once MAX-CUT is known to be complete, we show next that congestion games are also $\mathcal{PLS}$-complete, by transitiveness of polynomial-time reduction.

**Theorem 3.4.** All problems in $\mathcal{PLS}$ are polynomial-time reducible to the problem of finding a pure Nash equilibrium in congestion games.

*Proof.* Let $\mathcal{G} = (V, E)$ be a MAX-CUT instance with each edge $e \in E$ having capacity $w_e$.

We map each vertex $v \in V$ to a player. For each edge $e \in E$, we instantiate two resources $r_e$ and $\bar{r}_e$. Each player $v$ has two strategies of congestible resources: $s_v = \{r_e\}$ for all $r_e$ such that $v \in e$ and $\bar{s}_v = \{\bar{r}_e\}$ for all $\bar{r}_e$ such that $v \in e$. Thus, each resource associated with an edge $e$ is available only for the two players corresponding to its endpoints. Moreover, let $n = |V|$, since each player has 2 strategies, our $n$-matrix features $2^n$ strategy profiles, in perfect correspondence with the $2^{|V|}$ possible cuts[6] in $\mathcal{G}$.

Let $k$ be either $\bar{r}_e$ or $r_e$, the cost functions are defined as

$$c(f_k) = \begin{cases} 0 \text{ if } f_k = 1, \\ w_e \text{ else.} \end{cases}$$

Players aim to minimize their costs.

Fix a cut $(A, A^c)$. We establish the following bijection: a player $v$ who chooses its strategy $s_v$ corresponds to a vertex $v$ in partition $A$, and $v$ who chooses the strategy $\bar{s}_v$ corresponds then to a vertex $v$ in partition $A^c$.

Let $C(A, A^c)$ be the capacity of the cut. The game has potential function

$$\phi(\mathbf{s}) = \sum_{e \in E} w_e - C(A, A^c),$$

so maximizing the cut capacity is equivalent to minimizing Rosenthal function. Besides, one can check that under unilateral deviation, the change in $\phi$ is equal to (3.1), appropriately corresponding pure Nash equilibria – local minima in Rosenthal function – to local maxima in MAX-CUT. ∎

6. Previously, we stated that there are $2^{|V|-1}$ cuts for $\mathcal{G}$ because we were adjusting the count for symmetric cuts, e.g., $A = V$ is equivalent to $A = \emptyset$. One can check the game we are building preserves symmetry also, meaning the payoff for playing a particular strategy depends only on the other strategies employed, not on who is playing them. See Figure 3 and associated matrix for example.

In Figure 2, vertices colored in blue correspond to players choosing their respective strategy sets $s_v$, and vertices colored in pink correspond to players choosing their respective strategy sets $\bar{s}_v$. An edge $e$ between vertices of same color contributes with $w_e$ to its corresponding players' payoff each. Edges connecting vertices with different colors contribute with zero to their costs.

### 3.5 When is PNE tractable?

Up until here we discussed the nature of problems in $\mathcal{PLS}$ and argued that for complete problems in this class, a polynomial-time algorithm inexists or is yet to be seen. A natural question is whether or not we can find a PNE in polynomial time for particular instances. We already know the case for two-player zero-sum games can be solved by means of Linear Programming.

Linear Programming was first proved to solvable in polynomial time by (Khachiyan, 1979) by means of sequence of ellipsoids whose volume decreases at each iteration (see Bertsimas and Tsitsiklis, 1997; Capozzo, 2011). We say Linear Programming belongs to $\mathcal{FP}$ – FUNCTION POLYNOMIAL.

**Definition.** A problem $L$ belongs to $\mathcal{FP}$ if there is an algorithm $A$ that given any instance $x \in L$, $A$ finds $y$ such that $(x, y) \in R_L$ or reports that none exists with a number of steps of computation bounded by a polynomial in the length of $x$.

So the case for zero-sum games is tractable. It turns out for symmetric routing games – where all players share the same source and destination – the PNE can also be efficiently found by means of Linear Programming, reducing the problem to a *minimum-cost flow* (Fabrikant, Papadimitriou, and Talwar, 2004) with an amount of flow to be sent from source to sink equal to the number of agents playing the game.

The reduction is as follows: let $n$ be the number of players, for every edge $e$, instantiate $n$ parallel edges, each $i$-th edge with cost $c_e(i)$ and capacity equals to 1. So our game depicted in Figure 3 would become the min-cost flow problem in Figure 4.

The solution for min-cost flow would translate back to a PNE by using the number of parallels flows between each pair of vertices as the allocation of agents to the respective resource in the routing game.

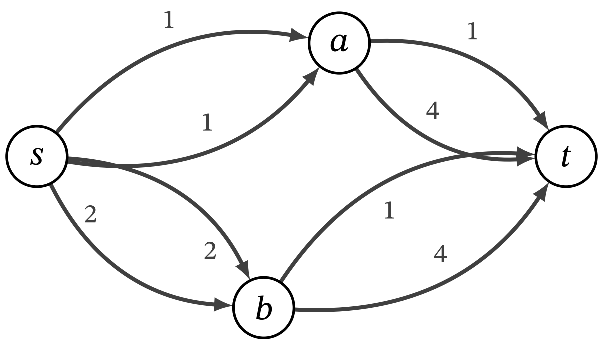

Figure 4: Min-cost flow for the atomic routing game in Figure 3.

So we know $\mathcal{FP} \subseteq \mathcal{PLS}$. Whether or not $\mathcal{FP}$ is a proper subset of $\mathcal{PLS}$ is still an open-problem in Computer Science. And even if a proof could show $\mathcal{PLS} = \mathcal{FP}$, a local search heuristic, in Algorithm 2, can potentially run in exponential time, so an efficient algorithm is yet to be seen, if it exists.

# 4 $\mathcal{PPAD}$ and Nash Equilibria

## 4.1 From Combinatorics...

Just like $\mathcal{PLS}$ class, another major class of interest which relates to the concepts of Game Theory we describe is the class $\mathcal{PPAD}$. We start with an important theorem of Combinatorics that surprisingly sheds light in "continuous" Mathematics properties and, by consequence, the proof of existence of equilibrium in general games.

**Definition.** An $(M-1)$-dimensional simplex, or $(M-1)$-simplex, is the convex hull of $M$ affinely independent points in the $\mathbb{R}^M$ space.

Simplices receive this name for they are, in a sense, the "simplest" figures of $(M-1)$ dimension. The 1-simplex is a line segment, 2-simplex is a triangle surface, the 3-simplex is a tetrahedron, and so on.

**Definition.** A *simplicial subdivision* of an $n$-dimensional simplex $T$ is a partition of $T$ into smaller simplices ("cells") such that any two cells share are either disjoint or share a face in a certain dimension.

**Definition.** A legal coloring of an $(M-1)$-simplex $T$ is an assignment of $M$ colors to each vertex of a simplicial division, such that vertices of $T$ receive different colors and points on each face of $T$ use only the colors of the vertices defining the respective face of $T$.

Notice the definition of legal coloring doesn't place restrictions on the coloring of internal vertices of the simplicial subdivision. As long as the vertices on the boundary of $T$ receive colors defined by the vertices of $T$ that define such boundary, the coloring is legal.

**Theorem 4.1.** (Sperner's Lemma) Any legal coloring of a simplicial division of an $n$-simplex has an odd number of $(n+1)$-colored cells.

*Proof.* Case $n = 1$:

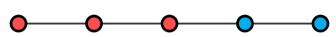

Figure 5: Legally colored simplicial division of a one-dimensional simplex.

The 1-simplex is a line segment, such as in Figure 5. Since its endpoints have different colors for the coloring to be legal, a path starting at one end and finishing at the other end must flip colors an odd number of times.

Case $n = 2$:

In this case, the 2-simplex is a triangle. Let us associate the colors yellow, red and blue to its vertices as illustrated in Figure 6.

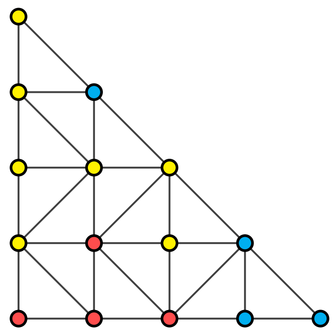

Figure 6: Legally colored simplicial division of a two-dimensional simplex.

Let $Y$ be the set of red-and-yellow-only triangles of the simplicial subdivision and $R$ be the set of tri-chromatic triangles. Let $t$ denote the number of red-and-yellow edges on the boundary of the simplex and $b$ the number of red-and-yellow edges in the interior of the simplex. Each triangle in $Y$ contributes with two red-and-yellow edges, and each triangle in $R$ contributes with one. This way, we count the internal edges twice - one for each triangle face:

$$2|Y| + |R| = t + 2b$$
$$|R| = t + 2(b - |Y|)$$

The legal coloring allows for red-and-yellow boundary edges only on the line between the vertices red and yellow of the 2-simplex, which reduces to the case already proved for $n = 1$, which implies $t$ is odd and so is the cardinality of $R$.

General case holds by induction on the number of dimensions:

Suppose the $(n-1)$-simplex to have an odd number of $n$-colored cells.

Let $F \subset S$ be the $(n-1)$-dimensional face of the $n$-simplex $S$, $F$ is a legally colored $(n-1)$-simplex with odd number of "rainbow" cells. Let $Y$ be the set of $n$-colored cells of

$S$ with colors indexed by $\{1, \ldots, n\}$ – such that exactly one color in $\{1, \ldots, n\}$ is used twice and all others once – again, $t$ and $b$ are the numbers of such faces on the boundary and the interior of $S$ respectively, and $R$ the set of rainbow cells – those that use all $n + 1$ colors.

The total number of faces with colors in $\{1, \ldots, n\}$.

$$2|Y| + |R| = t + 2b$$
$$|R| = t + 2(b - |Y|)$$

This reduces to the case for $(n - 1)$-simplex and, by hypothesis, $F$ has an odd number of $n$-colored cells, so $t$ is odd and so is $|R|$. ∎

Sperner's lemma poses a difficult problem for Computer Science on its own rights. Let us introduce an artificial vertex in the triangulation of Figure 6 such that we introduce another tri-chromatic triangle, depicted in Figure 7, on the underlying graph.

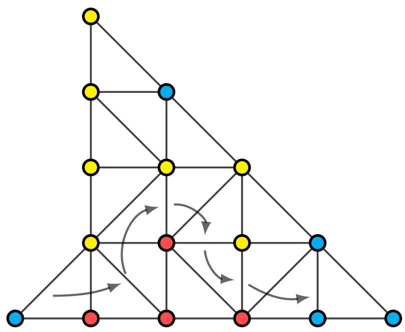

Figure 7: Traversal after introduction of an artificial tri-chromatic triangle.

We start off a path on the given graph, starting from the tri-chromatic triangle introduced through our artificial vertex. The traversal rule is simple: we can only travel through red-and-yellow "doors" with a yellow on the left-hand side.

It is straightforward from both the traversal rule and Sperner's lemma we cannot go backwards nor enter a cycle. In fact, the path should stop on another tri-chromatic triangle, guaranteed to exist. If we repeat the procedure starting from another arbitrarily chosen tri-chromatic triangle from inside the simplicial subdivision, again the path must end on another tri-chromatic triangle that is not the artificially created nor the endpoint of our first traversal.[7]

7. This is an equivalent way to make a construtive proof of Sperner's lemma for two-dimensional sim-

We construct a graph $G = (V, E)$ based on a Sperner's triangulation such that every cell is mapped to a vertex $v \in V$, with any two vertices $v$ and $u$ connected by a directed edge $(u, v)$ if and only if there's a red-yellow door between them – with the edge direction representing the direction of the traversal rule. We then have a graph with the property that every vertex has no more than one incident edge and no more than one outgoing edge.

At this point, we suppose an exponentially large graph whose vertices are each identified by a string of $n$ bits. The graph is defined by two circuits $P$ and $N$ and

$$\exists(u, v) \iff P(u) = v \land N(v) = u.$$

That is: $P$ is the circuit that points to a potential sucessor, and $N$ is the circuit that points to a potential predecessor. An edge can only exists between two vertices if the circuits agree.

The problem END-OF-LINE can now be stated.

> **End of Line.** Given an exponentially large graph with vertex set $\{0, 1\}^n$ with circuits $P$ and $N$. If $0^n$ is unbalanced, find another unbalanced node. Otherwise output $0^n$.

To be unbalanced means the vertex have an in-going edge, but not an out-going edge, and vice versa. Figure 8 shows an exponentially large graph on the vertex set $\{0, 1\}^4$.

By parity lemma in Combinatorics, we know the number of unbalanced nodes in a graph is even, so if $0^n$ is unbalanced then another unbalanced node must exist. Note the input length is polynomially bounded in $n$ – the $n$-bit string for the 0-th vertex – and checking all connections requires $\mathcal{O}(2^n)$ queries to the circuit.

**Lemma 4.2.** The number of vertices with odd degree in any graph is even.

*Proof.* Let $G = (V, E)$, denote the degree of a vertex by the function $d(\cdot)$. Each edge contributes with the degree of two vertices, so by summing the degree of all vertices:

$$\sum_{\forall v \in V} d(v) = 2|E|.$$

plices that can be stated as "all tri-chromatic triangles come in pairs, except for one – the one that pairs with the artificially introduced triangle." (Papadimitriou, 1994)

Which implies the number of odd degrees in the summation is even. ■

**Definition.** A problem $L \in \mathcal{PPAD}$ – POLYNOMIAL PARITY ARGUMENT FOR DIRECTED GRAPHS – if it is polynomial-time reducible to END-OF-LINE.

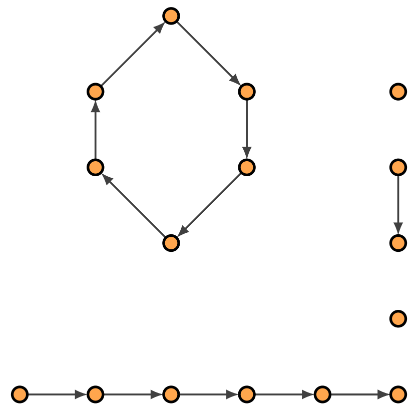

Figure 8: Exponentially large graph for END-OF-LINE with vertex set $\{0, 1\}^4$.

There is no known algorithm to solve END-OF-LINE in polynomial time in the size of the input, nor a proof that it is either possible or impossible to do so in polynomial time.

## 4.2 To topological statements

Sperner's lemma was proved in 1928 by Emmanuel Sperner and provided a simple way to prove a cornerstone theorem in Topology first proved by Luitzen Brouwer in 1911. We use Sperner's lemma to prove it here.

**Definition.** The *probability simplex* in $\mathbb{R}^m$ is the set of points $x = (x_1, \dots, x_m) \in \mathbb{R}^m$ such that $x_i \geqslant 0$, $\sum_{i=1}^{m} x_i = 1$.

The probability simplex is just the simplex as we defined earlier restricted to represent all families of probability distributions over an alphabet of size $m$. Because probabilities are nonnegative, this hyperplane lies on the first orthant with vertices at one unit distance from the origin. Figure 9 illustrates the probability for an alphabet of size 3. We use the probability simplex here to provide a natural way to speak about Mixed Nash equilibrium later.

**Theorem 4.3.** (Brouwer's fixed point) Let $\mathcal{B}$ be a convex and compact set – closed and bounded. Any continuous function $f : \mathcal{B} \mapsto \mathcal{B}$ have at least one fixed point, i.e., $f(x) = x$.

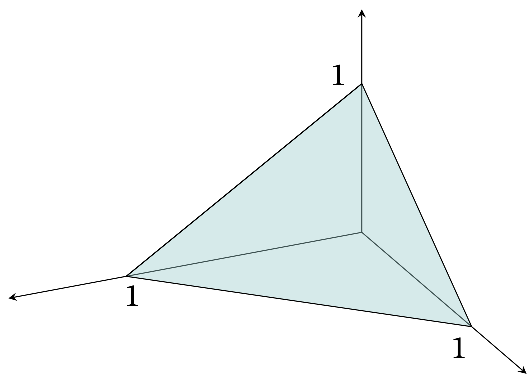


Figure 9: Two-dimensional probability simplex.

*Proof.* We prove the case here for the 2-simplex, the generality for any convex struture follows from homeomorphism in topological spaces, and the $n$-dimensional follows analogously using the general case for Sperner's Lemma in Theorem 4.1.

Let us assume $f : S \mapsto S$ is a continuous function where $S$ is the two-dimensional probability simplex and it is compact. Suppose further that there's no point $x = f(x)$. We define a coloring rule

$$c(x) = \begin{cases} 1 \text{ for } f(x)_1 < x_1, \\ 2 \text{ for } f(x)_1 \geqslant x_1 \text{ and } f(x)_2 < x_2, \\ 3 \text{ for } f(x)_1 \geqslant x_1,\, f(x)_2 \geqslant x_2 \text{ and } f(x)_3 < x_3 \end{cases}$$

As we assume a fixed point does not exist, there cannot be any point $x$ such that $x_1 \geqslant f(x)_1$, $x_2 \geqslant f(x)_2$, $x_3 \geqslant f(x)_3$.

Let us use blue, red and yellow to represent the colors 1, 2 and 3 respectively, as illustrated in Figure 10. Denote $\mathbf{p}_1 = (1, 0, 0)$, $\mathbf{p}_2 = (0, 1, 0)$, and $\mathbf{p}_3 = (0, 0, 1)$.

One can check that regardless the form of $f$ there is no possible point on the boundary between $\mathbf{p}_1$ and $\mathbf{p}_2$ able to be painted yellow, nor any point on the boundary between $\mathbf{p}_2$ and $\mathbf{p}_3$ able to receive blue, and no point on the boundary between $\mathbf{p}_1$ and $\mathbf{p}_3$ to be painted red, thus, the boundaries of the simplex satisfy the requirements for $S$ to be legally colored.

Define a simplicial subdivision in $S$ with infinitely many cells to be legally colored according to our painting rule $c(x)$. As the number of cells grows, the length of the edges

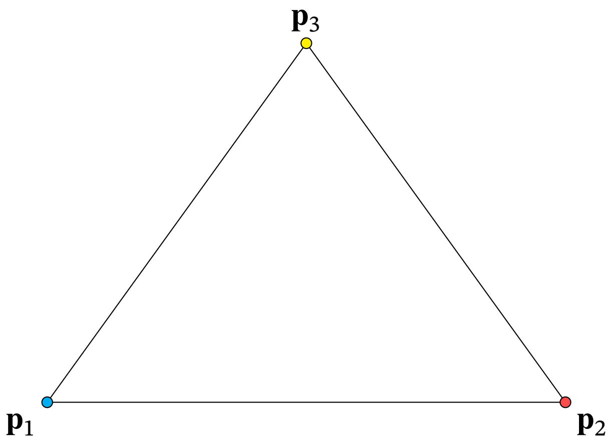

Figure 10: Two-dimensional probability simplex with colored vertices.

between any two points approaches zero, and there is a subsequence of triangles whose vertices all converge to some point $z$. By compactness of $S$, this point $z \in S$. Sperner's lemma implies there is at least one tri-chromatic triangle in the legally colored simplex, so blue vertices converge to $z \implies f(z)_1 \geqslant z_1$, red vertices converge to $z \implies f(z)_2 \geqslant z_2$ and yellow vertices converge to $z \implies f(z)_3 \geqslant z_3$. So $z = f(z)$, contradicting our assumption. ■

Since Sperner's lemma implies Brouwer's fixed point, the problem of finding a fixed point for a continuous function over a compact set, call it BROUWER, belongs to $\mathcal{PPAD}$.

## 4.3 General games and Nash equilibrium

Not all games have pure Nash equilibrium, as per our example with ROCK-PAPER-SCISSORS. Still, if we allow players to randomize over their strategy sets, an equilibrium is guaranteed to exist.

**Definition.** (Mixed Nash Equilibrium) In a cost-minimization game, a *Mixed Nash Equilibrium (MNE)* is an assignment of distributions $x_1, \dots, x_n$ over strategy sets $S_1, \dots, S_n$ such that

$$\mathop{\mathbf{E}}_{s_i \sim x_i, \mathbf{s}_{-i} \sim \mathbf{x}_{-i}}[u_i(\mathbf{s})] \leqslant \mathop{\mathbf{E}}_{s_i \sim x_i', \mathbf{s}_{-i} \sim \mathbf{x}_{-i}}[u_i(\mathbf{s})] \quad \forall\, i$$

and every unilateral deviation to probability distribution $x_i'$.

This definition generalizes the concept of equilibrium, and PNE is no different than a Mixed Nash Equilibrium where the support[8] of each player has cardinality equals to 1 each. The definition of MNE for profit-maximization game is analogous, changing the direction of the inequality.

Nash's theorem proves our Proposition (2.3.1) by constructing a continuous function over the cartesian products of players probability simplices, which is also a compact set.

For ilustration purposes, consider the profit-maximization Penalty game (Daskalakis, 2014):

| | LEFT | RIGHT |
|---|---|---|
| LEFT | $(-1, 1)$ | $(1, -1)$ |
| RIGHT | $(1, -1)$ | $(-1, 1)$ |

The kicker being the row-player and the goalkeeper the column player.

Each player has two strategies, so the probability simplices for each of them is a line segment $[0, 1]$. The cartesian product being the unit square $C = [0, 1]^2$. A continuous function $f : C \mapsto C$ tracks payoff increments over unilateral deviations for a given mixed strategy profile such that $\mathbf{E}[u_i(f(x))] \geqslant \mathbf{E}[u_i(x)]$ for some player $i$ with utility function $u_i(\cdot)$.

In Figure 11, the area shadowed in red are image points where the goalkeeper finds an incentive to increment the probability of diving to the right. The blue area, image points where the kicker finds an incentive to increment the probability of kicking to the left. The yellow area, image points where either the kicker finds an incentive to increment the probability of kicking to the right or the goalkeeper finds an incentive to increment the probability of diving to the left. The point $x^*$ represents the assignment of half of the probability for playing either left or right for both players and is the fixed point of $f$, so no color is assigned, meaning no player can increment its payoff under unilateral deviation.

**Theorem 4.4.** (Nash's Theorem) Every finite game has at least one *Mixed Nash equilibrium*.

*Proof.* Consider an $n$-player profit-maximization game, each player $i$ with strategy sets $S_i$

8. We call *support* the set of strategies associated with nonzero probabilities.

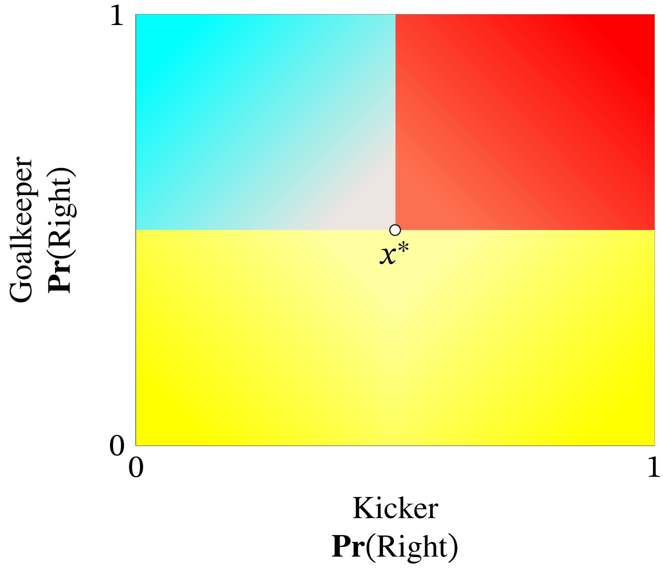


Figure 11: Compact set of interest for the Penalty game.

and utility function $u_i$. The relevant compact set is $C = \Delta_1 \times \cdots \times \Delta_n$, where $\Delta_i$ is the probability simplex of a player $i$.

We establish the functions $f_i(\cdot, \cdot)$ over points in the individual simplices $\Delta_i$:

$$f_i(x_i, \mathbf{x}_{-i}) = \underset{x_i' \in \Delta_i}{\operatorname{argmax}} \left[ \underset{s_i \sim x_i', \mathbf{s}_{-i} \sim \mathbf{x}_{-i}}{\mathbf{E}[u_i(\mathbf{s})]} - \|x_i' - x_i\|^2 \right]$$

The way to interpret this function (Roughgarden, 2016) is that the expectation part indicates a movement towards the best response strategy, whilst the second acts as a "penalty term" discouraging big changes in the vector of distributions. The function is concave in $x_i$ and thus local maxima exists.

Let $f = (f_1, \ldots, f_n)$, $f$ is continuous and map values from $C$ to $C$. Thus, by Brouwer's fixed point, it has at least one fixed point – a point where no player can increase its payoff by picking a different probability distribution to draw its strategies from, a Mixed Nash equilibrium. ■

Since BROUWER implies the existence of an MNE, pertinence of NASH to $\mathcal{PPAD}$ is proven. So, even though Nash's proof is derived from a topological statement, at its core, it is a combinatorial search problem. In fact, Lemke-Howson algorithm (Nisan et al., 2007), a known method of exponential running-time to find an MNE (out of the scope of this text) does so by exploiting directed paths on a graph defined by the support of probability distributions – and it even starts with an "artificial equilibrium" in a much similar manner to the artificial vertex for the path on the Sperner's triangulation.

Since the formulation of $\mathcal{PPAD}$ in (Papadimitriou, 1994) it was unknown whether NASH was $\mathcal{PPAD}$-complete. The positive result came years later on a celebrated proof by (Daskalakis, Goldberg, and Papadimitriou, 2008), which we do not reproduce here. This renders NASH a very difficult problem to solve and, by consequence, partially undermines its proposition as model of social behavior prediction – notice that games might have multiple equilibria, so even knowning a single equilibrium point, predictability of a system with multiple agents is not guaranteed. And beyond that: if a computer is unable to find an equilibrium, it is reasonable to think rational agents can't do either.

# 5 Final Considerations and Conclusions

## 5.1 Final Considerations

In this text, we presented the concept of Nash equilibrium, which is perhaps the most accepted concept of a "solution" to games in Game Theory. As equilibrium does not come from an algorithmic notion, but from a rather nonconstructive proof as per Nash's original works, Theoretical Computer Science took a natural lead into investigating fixed-point computation, giving rise to the topics studied in this monograph.

The birth of Game Theory and the formulation of the most basic form of a game, two-player zero-sum games, were proposed by John von Neumann and Oskar Morgenstern and published in 1928, with Theorem 2.7 being proved by means of fixed-point arguments (historical wonders discussed in Kjeldsen, 2001). Fastforward to the 1950s, George Dantzig presented his ideas on Linear Optimization Strong Duality and the SIMPLEX Algorithm, motivating von Neumann to prove Minimax again using arguments equivalent to the ones used here in (2.7). Even back then, Dantzig, 1951, conjectured Minimax was not only implied by strong duality, but in fact equivalent to it. This conjecture was proved only recently by (Adler, 2013).

Papadimitriou, 2001, argues that, since the 1980s, Computer Science has moved to connect more strongly with social sciences, largely motivated by the universality achieved by the Internet as an information repository, a common place operated by many parties with a multitude of economic interests "in varying relationships of collaboration and competition with each other." Thereby justifying the interaction between Computer Science and Game Theory and the underlying theory discussed herein, from the tractability of Linear Programming to the inadequacy of $\mathcal{NP}$-completeness to capture equilibrium complexity.

Although we do not approach such topics here, the theory presented in this monograph, both from algorithmic perspective and from game-theoretical ones contribute and improve upon interpretability of many applications in Sciences and Engineering, from channel capacity in Information Theory (see Cover and Thomas, 2006, Exercise 9.21) to machine learning models for online learning and boosting (Mohri, Rostamizadeh, and Talwalkar, 2018) and generative adversarial networks (Daskalakis et al., 2017; Daskalakis, 2018).

## 5.2 Future Works

There are related concepts of equilibria, considered to be computationally tractable, that we do not approach in this monograph. Namely, *correlated equilibrium* and *coarse correlated equilibrium*. Those cases involve interesting and different dynamics than Algorithm 3 and can also be used to prove Minimax (Theorem 2.7) by exploiting their convergence rules (Roughgarden, 2016) – as well as providing a more reasonable model on how agents learn by playing rather than finding exact equilibrium by solving linear programs. A natural path to expand upon this text is to work on the introduction of those concepts and algorithms.

Even more interesting, though, would be a deeper discussion about applications of the work discussed herein. Although motivated by applications, our approach focused mostly on complexity of computation, and enlarging the discussion with the use of the theory described here into domains of Machine Learning, Mechanism Design and Distributed Computation would signify a sizable improvement of the text and contribution towards the presentation of more digestible resources at the undergraduate and initial graduate levels.

# A Lagrange Multipliers

Lagrange Multipliers are a technique used in Optimization to find minima and maxima of a function subject to constraints. Consider the problem

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0 \end{aligned} \tag{A.1}$$

where the argument $x$ is a $D$-dimensional vector. The constraint $g(x) = 0$ specifies a region of feasible choices of $x$ and we seek to minimize $f$ as long as the argument $x$ lies in the constrained region. By projecting $g(x) = 0$ onto the surface levels of $f(x)$ one can see that the stationary points of $f(x)$ lie on surface levels tangent to the vicinity of $g(x) = 0$ – otherwise, for a given $x$, a new feasible point with better cost could be found by walking in the direction on the gradient at $x$ – thus gradient vectors $\nabla f(x)$ and $\nabla g(x)$ are antiparallel and must be equal up to a scaling factor

$$\nabla f(x) = -\lambda \nabla g(x), \tag{A.2}$$

where $\lambda \neq 0$ is called *Lagrange Multiplier*. Let

$$L(x, \lambda) \triangleq f(x) + \lambda g(x), \tag{A.3}$$

one can find optimal solutions to $f(x)$ subject to $g(x) = 0$ by computing the values of $x$ such that $\nabla_x L = 0$. Notice, also, derivating $L$ with respect to $\lambda$ returns the original constraint. Since $x$ is a $D$-dimensional vector, the derivatives should return a system of $D{+}1$ equations determining $x^*$ and $\lambda$. Typically, for discussions in the realm of Differential Calculus, the value of the Lagrange multiplier is not of interest, being the discussion on dual problems in Chapter 2 an otherwise application.

The Lagrange multipliers can be easily generalized for inequality constraints in the same vein discussed in Chapter 2, where sign constraints in the Lagrange multiplier are now present. The proof of Theorem 2.6 develops necessary conditions for optimality of a constrained optimization problem. The condition

$$(c - (p^*)'A)x^* = 0, \tag{A.4}$$

in particular, is known as *complementary slackness* and comes out as a corollary of strong duality in Theorem 2.5. In plain English, it states that either a decision variable for the primal is zero or its corresponding constraint in the dual is *active* – i.e., satisfied with equality. Conversely, for every *inactive* constraint in the primal, the associated variable in the dual assumes value zero. Together with primal and dual feasibility of $x^*$ and $p^*$, these conditions are known as *Karush-Kuhn-Tucker* conditions, for they were first stated by William Karush in his master's thesis (Karush, 1939), and later rediscovered independently by Harold W. Kuhn and Albert W. Tucker (Kuhn, 1950), (see also Kjeldsen, 2000, for historical wonders).

# B Linear Algebra

This section is only meant to review notation and basic operations of Linear Algebra, as no more than high-school Algebra is required to understand the concepts in this monograph.

**Definition.** A vector $x \in \mathbb{R}^n$ is the specification of a point $(x_1, \dots, x_n)$, were each $x_i$ specifies the distance from the origin in $i$-th coordinate.

We denote the zero vector, also known as *origin*, by $\mathbf{0}$. We abuse notation to denote $\mathbf{1}$ the vector were all entries are 1 unit from the origin. By vector we always mean a column vector. We use apostrophes to denote the transpose. For vectors are defined the operations of addition:

$$x + y = (x_1 + y_1, \dots, x_n + y_n),$$

where $x, y \in \mathbb{R}^n$; multiplication by scalars:

$$\lambda x = (\lambda x_1, \dots, \lambda x_n);$$

as well as the *inner-product*, also known as *dot-product*:

$$x'y = y'x = \sum_i x_i y_i,$$

where $x, y \in \mathbb{R}^n$, and it results in a scalar quantity.

**Definition.** The magnitude of a vector $x \in \mathbb{R}^n$ is denoted by $\|x\| = \sqrt{x'x} = \sqrt{\sum_i x_i^2}$.

**Definition.** A matrix $A \in \mathbb{R}^{m \times n}$ is an array of $n$ vectors of dimension $m$.

The operation of multiplication between a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $x \in \mathbb{R}^m$ is defined as the inner product between the column vectors of $A$ and $x$:

$$x'A = \begin{bmatrix} x'A_1 & x'A_2 & \cdots & x'A_n \end{bmatrix},$$

which results in a row vector of dimension $n$.

# C Asymptotic Notation

Operation counts for algorithms is estimated in terms of rate of growth of the number of arithmetic operations as function of the problem parameters. We use asymptotic notation to refer to orders of magnitude of the running time of an algorithm.

**Definition.** Let $f$ and $g$ be functions that map positive numbers to positive numbers.

i. We write $f(n) = \mathcal{O}(g(n))$ if there exist positive number $n_0$ and positive constant $c$ such that $f(n) \leqslant cg(n)$ for all $n \geqslant n_0$.

ii. We write $f(n) = \Omega(g(n))$ if there exist positive number $n_0$ and positive constant $c$ such that $f(n) \geqslant cg(n)$ for all $n \geqslant n_0$.

iii. We write $f(n) = \Theta(g(n))$ if there exist positive number $n_0$ and positive constants $c_1$ and $c_2$ such that $c_1 g(n) \leqslant f(n) \leqslant c_2 g(n)$ for all $n \geqslant n_0$.

# D Definitions from Topology and Real Analysis

For the purposes of this monograph, a *compact* set can be understood as a set that is both closed and bounded.

**Definition.** A subset $S \subset \mathbb{R}^n$ is bounded if, and only if, there exist finite $r$ and ball $B = (\mathbf{0}, r)$ such that $S \subseteq B$.

**Definition.** Let $x$ be a point in $\mathbb{R}^n$. The set of all points in an arbitrary nonzero radius from $x$ is a *neighborhood* of $x$.

**Definition.** Let $S$ be a subset of $\mathbb{R}^n$. A point $x \in \mathbb{R}^n$ is a *limit point* of $S$ if every neighborhood of $x$ contains at least one point of $S$ different from $x$.

The set $S$ is said to be *closed* if it contains all of its limit points.

Intuitively, a set $S$ is closed if it contains all of its boundaries. Linear Programming, defined in Chapter 2, can only defined for compact polyhedra – otherwise, for any point $x$ inside the polyhedron, we could add an infinitesimally small quantity $|\epsilon| > 0$ to some of its components in order to optimize for the objective function without violating the constraints, such that the objective function nevers attains its infimum or supremum. A more detailed discussion requires the application of Weierstrass' Extreme Value Theorem, which states that for every continuous real-valued function over a compact set there exist at least a minimum and a maximum for the said function, but we leave it out of the scope of this monograph.

**Definition.** (*Homeomorphism*) Two sets $S$ and $R$ are said to be *homeomorphic* to each other if, and only if, there is a continuous and invertible bijective function $g : S \mapsto R$, with $\mathrm{g}^{-1}$ being continuous also.

Homeomorphism is an important concept in Topology, as it establishes a notion of equivalence between geometrical spaces, or more generally *topological spaces*. For instance, some set with a "hole" inside can be proven to be homeomorphic to a torus, so any property that is true for a torus holds for the initial figure.[9] Likewise, homeomorphism plays a role in the generalization of Theorem 4.3 to convex structures, for the reason that

9. There's a well-known joke among mathematicians that a topologist can't discern a donut from a coffee mug.

any convex $D$-dimensional structure is homeomorphic to a $D$-simplex, thus proving the existence of a fixed point for any continuous function $f : S \mapsto S$ where $S$ is convex.

We refer to (Tao, 2006; Rudin, 1976) for textbooks on Analysis, and (Theodore W. Gamelin, 1999) for a treatment in Topology.

# E Graph Theory

Although graphs are not the main object of study in this monograph, Graph Theory underlies many concepts approached herein to the point we use graph-theoretic jargons as a second nature throughout. Therefore, some definitions are presented in this section in order clarify any jargon used in the text.

**Definition.** A graph $G = (V, E)$ is a tuple, where $V$ is the set of vertices and $E$ is the set of edges connecting pair of vertices $u, v \in V$.

If the edges have directions associated, each edge $e$ is a tuple $(u, v)$ such that the direction is from $u$ to $v$.

**Definition.** A *walk* is any arbitrary sequence of vertices $(v_1, v_2, \dots, v_n)$ such that there exists edges between each pair of vertices $v_i, v_{i+1}$.

**Definition.** A *path* is a walk such that no vertice is present in the sequence more than once.

**Definition.** A *cycle* is a sequence of vertices $(v_1, v_2, \dots, v_{n-1}, v_n)$ such that the subsequence $(v_1, \dots, v_{n-1})$ is a path and $v_1 = v_n$.

**Definition.** The *degree* of a vertice $v \in V$ is the number of edges incident to $v$.

# References


Adler, Ilan. 2013. "The equivalence of linear programs and zero-sum games." *International Journal of Game Theory* 42 (1): 165–177.

Bertsimas, Dimitris, and John Tsitsiklis. 1997. *Introduction to Linear Optimization.* 1st. Athena Scientific. ISBN: 1886529191.

Borzechowski, Michaela. 2016. "The Complexity Class Polynomial Local Search and PLS-complete problems," Institut für Informatik, Freien Universität Berlin.

Capozzo, Samantha. 2011. "A New Proof of the Ellipsoid Algorithm," College of Letters and Sciences, UC Davis.

Cover, Thomas M., and Joy A. Thomas. 2006. *Elements of Information Theory.* USA: Wiley-Interscience. ISBN: 0471241954.

Dantzig, George B. 1951. "A proof of the equivalence of the programming problem and the game problem." In *Activity Analysis of Production and Allocation,* 330–335. John Wiley & Sons, Inc.

Daskalakis, Constantinos. 2014. "PPAD." Chap. 22 in *6.890 Algorithmic Lower Bounds: Fun with Hardness Proofs,* edited by Erik Demaine. License: Creative Commons BY-NC-SA. https://youtu.be/TUbfCY_8Dzs.

———. 2018. "Equilibria, Fixed Points, and Computational Complexity - Nevanlinna Prize Lecture." In *Proceedings of International Congress of Mathematicians,* 147–210. World Scientific Publishing Company.

Daskalakis, Constantinos, Paul Goldberg, and Christos Papadimitriou. 2008. "The Complexity of Computing a Nash Equilibrium."

Daskalakis, Constantinos, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. 2017. *Training GANs with Optimism.* https://doi.org/10.48550/arXiv.1711.00141.

Fabrikant, Alex, Christos Papadimitriou, and Kunal Talwar. 2004. "The Complexity of Pure Nash Equilibria." In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing,* 604–612. STOC '04. Chicago, IL, USA: Association for Computing Machinery. ISBN: 1581138520. https://doi.org/10.1145/1007352.1007445.

Farkas, Julius. 1902. *Journal für die reine und angewandte Mathematik (Crelles Journal),* no. 124, 1–27. https://doi.org/doi:10.1515/crll.1902.124.1.

Garey, Michael R., and David S. Johnson. 1990. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* USA: W. H. Freeman & Co. ISBN: 0716710455.

Karush, William. 1939. "Minima of functions of several variables with inequalities as side conditions.," University of Chicago, Department of Mathematics, December 1939.

Khachiyan, Leonid Genrikhovich. 1979. "A Polynomial Algorithm in Linear Programming." *Soviet Mathematics Doklady,* no. 20, 191–194.

Kjeldsen, Tinne Hoff. 2000. "A Contextualized Historical Analysis of the Kuhn–Tucker Theorem in Nonlinear Programming: The Impact of World War II." *Historia Mathematica* 27 (4): 331–361. ISSN: 0315-0860. https://doi.org/https://doi.org/10.1006/hmat.2000.2289.

———. 2001. "John von Neumann's Conception of the Minimax Theorem: A Journey Through Different Mathematical Contexts." *Archive for History of Exact Sciences* 56 (1): 39–68. ISSN: 00039519, 14320657. http://www.jstor.org/stable/41134130.

Kothari, C.R. 2004. *Research Methodology: Methods and Techniques.* New Age International (P) Limited. ISBN: 9788122415223.

Kuhn, A. W., H. W.; Tucker. 1950. "Nonlinear Programming." In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability,* edited by Berkeley University of California Press, 481–492.

Lawler, Eugene L. 2011. *Combinatorial Optimization: Networks and Matroids.* Dover Publications. ISBN: 0486414531.

Leyton-Brown, Kevin, and Yoav Shoham. 2008. *Essentials of Game Theory: A Concise, Multidisciplinary Introduction.* 1st. Morgan & Claypool Publishers. ISBN: 1598295934.

Mangasarian, O.L., and H. Stone. 1964. "Two-person nonzero-sum games and quadratic programming." *Journal of Mathematical Analysis and Applications* 9 (3): 348–355. ISSN: 0022-247X. https://doi.org/10.1016/0022-247X(64)90021-6.

Mankiw, N. Gregory. 2016. *Principles of Microeconomics.* CENGAGE Learning Custom Publishing. ISBN: 9781305971493.

Mohri, Mehryar, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. *Foundations of Machine Learning.* 2nd. The MIT Press. ISBN: 0262039400.

Nisan, Noam, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani, eds. 2007. *Algorithmic Game Theory.* USA: Cambridge University Press, September. ISBN: 0521872820.

OUP, Oxford University Press, ed. 2018. "Textbook." In *Oxford Learner's Dictionary.* https://www.oxfordlearnersdictionaries.com/definition/english/textbook_1.

Papadimitriou, Christos H. 1994. "On the complexity of the parity argument and other inefficient proofs of existence." *Journal of Computer and System Sciences* 48 (3): 498–532. ISSN: 0022-0000. https://doi.org/10.1016/S0022-0000(05)80063-7.

———. 2001. "Algorithms, Games, and The Internet." In *STOC '01: Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing.* Hersonissos, Greece: Association for Computing Machinery. ISBN: 1581133499.

Papadimitriou, Christos H., and Kenneth Steiglitz. 1998. *Combinatorial Optimization: Algorithms and Complexity.* Dover Publications. ISBN: 978-0-486-40258-1.

Rosenthal, Robert W. 1973. "A class of games possessing pure-strategy Nash equilibria." *International Journal of Game Theory* 2 (1): 65–67. https://doi.org/10.1007/BF01737559.

Roughgarden, Tim. 2016. *Twenty Lectures on Algorithmic Game Theory.* 1st. USA: Cambridge University Press. ISBN: 131662479X.

Rudin, Walter. 1976. *Principles of Mathematical Analysis.* McGraw Hill. ISBN: 978-0070542358.

Schäffer, Alejandro A., and Mihalis Yannakakis. 1991. "Simple Local Search Problems That Are Hard to Solve." *SIAM J. Comput.* (USA) 20, no. 1 (February): 56–87. ISSN: 0097-5397. https://doi.org/10.1137/0220004.

Schrijver, Alexander. 2005. "On the History of Combinatorial Optimization (Till 1960)." In *Discrete Optimization,* edited by K. Aardal, G.L. Nemhauser, and R. Weismantel, 12:1–68. Handbooks in Operations Research and Management Science. Elsevier. https://doi.org/10.1016/S0927-0507(05)12001-5.

Sipser, Michael. 2012. *Introduction to the Theory of Computation.* 3rd. Cengage Learning. ISBN: 978-1-133-18779-0.

Stengel, Bernhard von. 2022. *Zero-Sum Games and Linear Programming Duality.* https://doi.org/10.48550/arxiv.2205.11196.

Tao, Terence. 2006. *Analysis I.* Hindustan Book Agency/Springer Science. ISBN: 978-981-10-1789-6.

Theodore W. Gamelin, Robert E. Greene. 1999. *Introduction to Topology.* Dover Publications. ISBN: 978-0486406800.