Algorithmic Game Theory
FAU WS 2022/23

Yiannis Giannakopoulos*

Last update: March 6, 2023

Summary

1 Introduction **4**
1.1 Normal Form Games 4
1.1.1 Bimatrix Games 4
1.2 Basic Solution Concepts 5
1.2.1 Payoff-Maximization Games 5
1.3 Examples of Games 6

2 Congestion Games **7**
2.1 Potential Games 7
2.2 Congestion Games 7
2.3 Price of Anarchy and Stability 10
2.4 Computation of Equilibria 13
2.4.1 Input Representation 13
2.4.2 Better-Response Dynamics 14
2.4.3 Singleton Games 15
2.4.4 Approximate Equilibria and Symmetric Games 15
2.4.5 Network Symmetric Games 17

3 Computation of Pure Nash Equilibria **19**
3.1 Local Search and the Class PLS 19
3.1.1 Better-Response Graphs 19
3.1.2 Maximum Cuts 19
3.1.3 Local Search Problems 19
3.2 PLS-completeness 21

4 Mixed Nash Equilibria **23**
4.1 Bimatrix Games 23
4.2 Zero-Sum Games 23
4.3 Fictitious Play 25
4.4 Nash's Theorem 26

*Department of Data Science, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU).
Email: `yiannis.giannakopoulos@fau.de`, Web: `yiannisgiannakopoulos.com`

5 Correlated Equilibria **28**
5.1 Definitions 28
5.2 No-Regret Learning 29
5.2.1 Multiplicative Weights Update 31
5.3 No-Regret Game Dynamics 33
5.4 Swap Regret 34

Acknowledgements

These notes have been developed around my teaching of Algorithmic Game Theory in TU Munich (2018–2020) and FAU Erlangen-Nürnberg (2021–2023). I am deeply grateful to all my students for the opportunity; their engagement and our interactions have shaped and advanced my understanding of the topics in a profound way.

I also want to thank many of my fellow researchers in the field of Algorithmic Game Theory, the already existing teaching material of whom has inspired many parts of these lecture notes. In some cases, I have even unashamedly directly used their exposition. Those include: Costis Daskalakis, Paul Dütting, Martin Hoefer, Thomas Kesselheim, Pieter Kleer, Max Klimm, and Tim Roughgarden. Of course, all mistakes are mine.

In particular, I am indebted to Tim Roughgarden for his inspiring lecture notes, videos, and textbooks, all of which have had a significant effect on my teaching. Some parts of these notes are heavily based on his work.

For pedagogical reasons, bibliographical references are kept to a minimum throughout the text. For completeness, I have set up an accompanying web page, where detailed citations and further literature pointers are provided.

1 Introduction

1.1 Normal Form Games

A (finite, cost-minimization) *game* $\mathcal{G} = (N, \{S_i\}_{i \in N}, \{C_i\}_{i \in N})$ consists of

- a finite set of *players* $N = [n] \coloneqq \{1, 2, \dots, n\}$
- for each player $i \in N$, a (nonempty) finite set S_i of *strategies* (or actions)
- for each player $i \in N$, a *cost* function $C_i : S_1 \times \dots \times S_n \longrightarrow \mathbb{R}$.

An n-tuple $\boldsymbol{s} = (s_1, \dots, s_n) \in \boldsymbol{S} = S_1 \times \dots S_n$ of strategies, one for each player $i \in N$, is called a *strategy profile* or *outcome* of $\mathcal{G}$. The *social cost* $C(\boldsymbol{s})$ of a strategy profile $\boldsymbol{s}$ is defined as the sum of the players' costs:

$$C(\boldsymbol{s}) \coloneqq \sum_{i \in N} C_i(\boldsymbol{s}).$$

A game $\mathcal{G}$ is called *zero-sum* if all outcomes have zero social cost; i.e., $\sum_{i \in N} C_i(\boldsymbol{s}) = 0$ for all $\boldsymbol{s} \in \boldsymbol{S}$.

A *mixed* strategy σ_i of player i is a probability distribution over her (pure) strategies S_i. Formally, the set of mixed strategies of a player $i \in N$ is defined as $\Delta(S_i)$, where for an arbitrary finite set A, $\Delta(A)$ denotes the the $(|A| - 1)$-dimensional simplex

$$\Delta(A) \coloneqq \left\{ p \in [0,1]^A \;\middle|\; \sum_{a \in A} p(a) = 1 \right\}.$$

Then, a *mixed* strategy profile of $\mathcal{G}$ is a *product* distribution $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_n) \in \Delta(S_1) \times \dots \times \Delta(S_n)$ over the set $\boldsymbol{S}$ of pure strategy profiles.

Notation For any vector $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$ and positive integer $i \leq n$, we use the standard game-theoretic notation

$$\boldsymbol{x}_{-i} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$$

for the $(n-1)$–dimensional vector that we get from $\boldsymbol{x}$ if we remove its i-th coordinate. Then, for any $y \in \mathbb{R}$, we can use the following shorthand notation for the vector that results from $\boldsymbol{x}$ if we replace its i-th coordinate with y:

$$(y, \boldsymbol{x}_{-i}) = (x_1, \dots, x_{i-1}, y, x_{i+1}, \dots, x_n).$$

In particular, $\boldsymbol{x} = (x_i, \boldsymbol{x}_{-i})$.

1.1.1 Bimatrix Games

For the special case of $n = 2$, it is common to represent two-player games through their *cost matrix* (see, for example, the game examples discussed in the following Section 1.2). The number of rows of such a matrix is equal to the cardinality of the strategy set S_1 of player 1 (called the *row player*), and the number of its columns to the cardinality of the strategy set S_2 of player 2 (called the *column player*). Each entry of this matrix, is a tuple: its first component gives the corresponding cost of player 1 and its second component that of player 2. Formally, if we enumerate the strategy sets $S_1 = \{s_{1,1}, \dots, s_{1,m_1}\}$, $S_2 = \{s_{2,1}, \dots, s_{2,m_2}\}$, then the (i, j)-th entry of the cost matrix is $\left(C_1(s_{1,i}, s_{2,j}), C_2(s_{1,i}, s_{2,j})\right)$.

Equivalently, such a game can be also represented by two matrices $A, B \in \mathbb{R}^{m_1 \times m_2}$, where $A = \left(C_1(s_{1,i}, s_{2,j})\right)$ is the cost matrix of the row player and $B = \left(C_2(s_{1,i}, s_{2,j})\right)$ that of the column player. For that reason, two-player games are also called *bimatrix*, since they can be fully described by the tuple (A, B). Notice that two-player, zero-sum games have the special form $(A, -A)$, and thus they can be fully determined by a single matrix A.

1.2 Basic Solution Concepts

Fix some game $\mathcal{G} = (N, \{S_i\}_{i\in N}, \{C_i\}_{i\in N})$. A strategy $s_i \in S_i$ will be called (weakly) *dominant* for player i if it minimizes her cost, no matter the actions of the other players. Formally, for any $\boldsymbol{s}'_{-i} \in \boldsymbol{S}_{-i}$ and all $s'_i \in S_i$:

$$C_i(s_i, \boldsymbol{s}'_{-i}) \leq C_i(s'_i, \boldsymbol{s}'_{-i}). \tag{1}$$

A profile $\boldsymbol{s} = (s_1, \dots, s_n)$ is a *dominant strategy equilibrium* of $\mathcal{G}$, if s_i is a dominant strategy of player i, for all players $i \in N$.

A profile $\boldsymbol{s} = (s_1, \dots, s_n)$ is a *pure Nash equilibrium (PNE)* of $\mathcal{G}$ if no player has an incentive to unilaterally deviate from it. Formally, for any player $i \in N$ and all strategies $s'_i \in S_i$,

$$C_i(\boldsymbol{s}) \leq C_i(s'_i, \boldsymbol{s}_{-i}). \tag{2}$$

Similarly, considering the *expected* costs of the players, a *mixed* strategy profile $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_n)$ will be called *(mixed) Nash equilibrium (NE)* if, for any player $i \in N$ and all mixed strategies $\sigma'_i \in \Delta(S_i)$:

$$C_i(\boldsymbol{\sigma}) \leq C_i(\sigma'_i, \boldsymbol{\sigma}_{-i}), \tag{3}$$

where we are making use of the shorthand notation

$$C_i(\boldsymbol{\sigma}) \coloneqq \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\sigma}}\left[C_i(\boldsymbol{s})\right] \tag{4}$$

for the expected cost of a player i at a mixed outcome $\boldsymbol{\sigma}$. Formally, since the components of $\boldsymbol{s}$ in (4) are independent random variables, the expected costs are given by:

$$C_i(\boldsymbol{\sigma}) = \sum_{\boldsymbol{s}\in\boldsymbol{S}}\left(\prod_{j\in N}\sigma_j(s_j)\right)C_i(\boldsymbol{s}) = \sum_{s_1\in S_1}\sum_{s_2\in S_2}\cdots\sum_{s_n\in S_n}\sigma_1(s_1)\cdots\sigma_n(s_n)\cdot C_i(s_1,\dots,s_n).$$

It can be shown (see tutorials) that conditions (3) can be (without loss) simplified to consider only pure-strategy deviations.

It is not difficult to see (verify this on your own!) that the following hierarchy of the sets of equilibria holds for any game:

$$\{\text{dominant strategy}\} \subseteq \{\text{pure Nash}\} \subseteq \{\text{mixed Nash}\}.$$

In Section 4.4 we will prove that:

Theorem 1.1 (Nash [1950]). *Every finite, normal-form game has at least one mixed Nash equilibrium.*

1.2.1 Payoff-Maximization Games

So far we have modelled the players as cost-minimizers: their own, selfish goal is to choose a strategy with the *smallest* possible cost. However, many applications naturally call for a different modelling, where at every outcome of the game there is an actual *gain* to be received by each player; now, players are selfishly trying to *maximize* their own benefit, instead of minimizing their costs. For example, this is the case with the BoS game described below (Example 1.2). In such games, we usually use *payoff* (or *utility*) functions $u_i : \boldsymbol{S} \longrightarrow \mathbb{R}$, which can be simply viewed as negative costs, i.e., $u_i(\boldsymbol{s}) = -C_i(\boldsymbol{s})$ for any strategy profile $\boldsymbol{s}$. Thus, formally we can define a *payoff-maximization game* $\mathcal{G} = (N, \{S_i\}_{i\in N}, \{u_i\}_{i\in N})$ as simply being a standard, *cost-minimization* game $\mathcal{G} = (N, \{S_i\}_{i\in N}, \{-u_i\}_{i\in N})$.

All solution concepts described above, can be adapted in the natural way to payoff-maximization games, by simply switching the direction of the inequality in their defining conditions (that is, (1), (2) and (3)). For example, condition (2) for the pure Nash equilibrium now becomes

$$u_i(\boldsymbol{s}) \geq u_i(s'_i, \boldsymbol{s}_{-i}).$$

1.3 Examples of Games

Example 1.1 (Prisoners' Dilemma)**.** The following game has a unique (verify!) dominant strategy equilibrium, namely (*confess*, *confess*).

		Prisoner 2	
		confess	*silent*
Prisoner 1	*confess*	5, 5	1, 10
	silent	10, 1	2, 2

Table 1: Prisoner's Dilemma

Example 1.2 (Bach or Stravinsky)**.** The following payoff-maximization game has *no* dominant strategy equilibria (check this!). However, it has two pure Nash equilibria, namely (B, B) and (S, S). (It also has one additional *mixed* Nash equilibrium; can you find it?)

		Player 2	
		B	*S*
Player 1	*Bach (B)*	10, 7	5, 5
	Stravinsky (S)	1, 1	7, 10

Table 2: Bach or Stravinsky (BoS)

Example 1.3 (Rock-Paper-Scissors)**.** The following (zero-sum) payoff-maximization game has no dominant strategy equilibria, neither PNEs (check this!). However, it has a *mixed* NE, namely the profile $\left(\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right), \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)\right)$ which corresponds to both players randomizing uniformly at random over their pure strategies. (Can you prove that this is actually the *unique* NE of the game?)

	R	*P*	*S*
Rock (R)	0, 0	−1, 1	1, −1
Paper (P)	1, −1	0, 0	−1, 1
Scissors (S)	−1, 1	1, −1	0, 0

Table 3: Rock-Paper-Scissors

2 Congestion Games

2.1 Potential Games

Definition 2.1 (Potential Game, Monderer and Shapley [1996]). A game $\mathcal{G} = (N, \{S_i\}_{i\in N}, \{C_i\}_{i\in N})$ is an *(exact) potential game*, if there exists a function $\Phi : \mathbf{S} \longrightarrow \mathbb{R}$ such that

$$C_i(s_i', \mathbf{s}_{-i}) - C_i(\mathbf{s}) = \Phi(s_i', \mathbf{s}_{-i}) - \Phi(\mathbf{s}), \tag{5}$$

for any strategy profile $\mathbf{s} \in \mathbf{S}$, any player $i \in N$, and any deviation $s_i' \in S_i$. Such a function Φ, that satisfies (5), is called an *(exact) potential* of game $\mathcal{G}$.

Example 2.1. The Prisoner's Dilemma game (see Table 1) is a potential game. A potential function for it can be seen in Table 4.

		Prisoner 2	
		confess	*silent*
Prisoner 1	*confess*	0	5
	silent	5	6

Table 4: A potential function for Prisoner's Dilemma

Theorem 2.1. *Every potential game has at least one pure Nash equilibrium.*

Proof. Fix a (finite, cost-minimization) potential game $\mathcal{G} = (N, \{S_i\}_{i\in N}, \{C_i\}_{i\in N})$, along with some potential function Φ for $\mathcal{G}$. Let

$$\mathbf{s}^* \in \operatorname*{argmin}_{\mathbf{s}\in\mathbf{S}} \Phi(\mathbf{s})$$

be a minimizer of the potential function; this is well-defined, since the space $\mathbf{S}$ of all feasible strategy profiles is finite. We will prove that $\mathbf{s}^*$ is a pure Nash equilibrium of $\mathcal{G}$.

Indeed, for any player i and all strategies $s_i' \in S_i$ we have

$$C_i(s_i', \mathbf{s}_{-i}^*) - C_i(\mathbf{s}^*) = \Phi(s_i', \mathbf{s}_{-i}^*) - \Phi(\mathbf{s}^*) \geq 0, \tag{6}$$

which is equivalent to the pure Nash equilibrium condition

$$C_i(\mathbf{s}^*) \leq C_i(s_i', \mathbf{s}_{-i}^*).$$

The first equality in (6) holds because Φ is a potential, and the last inequality because $\mathbf{s}^*$ is a minimizer of Φ. □

2.2 Congestion Games

A *congestion game* $\mathcal{G} = (N, E, \{S_i\}_{i\in N}, \{c_e\}_{e\in E})$ is a game $(N, \{S_i\}_{i\in N}, \{C_i\}_{i\in N})$ with the following additional structure:

- A finite set of *resources* (or *facilities*) E. Each resource $e \in E$ has a *nondecreasing* cost function $c_e : \mathbb{N} \longrightarrow \mathbb{R}_{\geq 0}$, representing the *congestion* induced on it, as a function of the number of players that use it.

- The strategies of the players are sets of resources; i.e., $S_i \subseteq 2^E$ for all $i \in N$.

- The cost of player i on a given strategy profile $\boldsymbol{s} \in \boldsymbol{S}$ is the total congestion experienced by i (on her set s_i of selected resources), that is

$$C_i(\boldsymbol{s}) = \sum_{e \in s_i} c_e(n_e(\boldsymbol{s})),$$

where $n_e(\boldsymbol{s}) = |\{i \mid e \in s_i\}|$ denotes the number of players that use resource e in $\boldsymbol{s}$.

Then, the social cost of an outcome $\boldsymbol{s}$ of a congestion game is the total congestion experienced by all players, i.e.

$$C(\boldsymbol{s}) = \sum_{i=1}^{n} C_i(\boldsymbol{s}) = \sum_{i=1}^{n} \sum_{e \in s_i} c_e(n_e(\boldsymbol{s})) = \sum_{e \in E} n_e(\boldsymbol{s}) \cdot c_e(n_e(\boldsymbol{s})).$$

For congestion games, condition (2) in the definition of a pure Nash equilibrium can be equivalent rewritten as

$$\begin{aligned}\sum_{e \in s_i} c_e(n_e(\boldsymbol{s})) &\leq \sum_{e \in s_i' \cap s_i} c_e\left(n_e(s_i', \boldsymbol{s}_{-i})\right) + \sum_{e \in s_i' \setminus s_i} c_e\left(n_e(s_i', \boldsymbol{s}_{-i})\right) \\ &= \sum_{e \in s_i' \cap s_i} c_e\left(n_e(\boldsymbol{s})\right) + \sum_{e \in s_i' \setminus s_i} c_e\left(n_e(\boldsymbol{s}) + 1\right). \qquad (7)\end{aligned}$$

The following special classes of congestion games are of particular interest:

Network congestion games also known as *(atomic) routing games*: these are congestion games with the following additional structure:

- A *directed* graph $G = (V, E)$
- Each player has a special pair of nodes $o_i, d_i \in V$ (called *origin* and *destination*, respectively)
- The strategies of a player are *all* the different ways to travel from her origin to her destination; formally,

$$S_i = \{\pi \mid \pi \text{ is a (simple) } o_i \longrightarrow d_i \text{ path in } G\}.$$

Polynomial congestion game A congestion game is called *polynomial of degree d*, for d nonnegative integer, if all its cost functions are polynomials of degree (at most) d with nonnegative coefficients. Formally, for every $e \in E$:

$$c_e \in \mathcal{C}_d \coloneqq \{x \mapsto \sum_{j=0}^{d} a_j x^j \mid a_j \in \mathbb{R}_{\geq 0} \quad \forall j = 0, \dots, d\}.$$

Polynomial congestion games of degree 1, i.e. having affine cost functions $\mathcal{C}_1$, will be called *linear* congestion games.

Singleton congestion games These are congestion games where the strategies of the players are simply resources (as opposed to *sets* of resource, in the general case). Formally, in a singleton congestion game it is $|s_i| = 1$ for all $i \in N$ and $s_i \in S_i$.

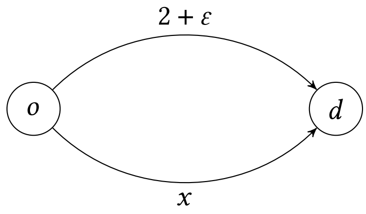

Figure 1: A Pigou network. There are two players, both of which want to travel from o to d. The social optimum is to send them from different paths, for a total cost of $2+\varepsilon+1=3+\varepsilon$. Going both through the bottom link, for a total cost of $2+2=4$, is the only (pure Nash) equilibrium of the game.

Symmetric congestion games These are congestion games where all players have the same set of available actions, i.e., $S_i = S_j$ for all $i, j \in N$.

Example 2.2 (Pigou network)**.** In Figure 1 you can see an example of a congestion game which is simultaneously linear, symmetric, network and singleton. There are two players, both of which want to travel from o to d. There are only two parallel links (with end-points o, d), one having a constant cost function of $c(x) = 2+\varepsilon$ (where ε is an arbitrary constant with $0 < \varepsilon < 1$) and the other a linear $c(x) = x$. Therefore, there are only two strategies available to both players: "*Up*" or "*Down*"

The (bimatrix) normal-form representation of this game can be seen in Table 5. One can easily see that (*Down*, *Down*) is the *only* pure Nash equilibrium of the game; as a matter of fact, choosing the bottom link is a strongly dominant strategy for both players.

	Up	*Down*
Up	$2+\varepsilon, 2+\varepsilon$	$2+\varepsilon, 1$
Down	$1, 2+\varepsilon$	$2, 2$

Table 5: Bimatrix representation of the Pigou congestion game of Figure 1.

Theorem 2.2 (Rosenthal [1973])**.** *Every congestion game has at least one pure Nash equilibrium.*

Proof. Due to Theorem 2.1, it is enough to prove that all congestion games are potential games.

Fix a congestion game, and for all strategy profiles $\boldsymbol{s}$ define

$$\Phi(\boldsymbol{s}) = \sum_{e\in E} \sum_{j=1}^{n_e(\boldsymbol{s})} c_e(j). \tag{8}$$

We will show that Φ is a potential function for our congestion game. The particular choice of potential in (8) is known as *Rosenthal's potential* function.

Fix a strategy profile $\boldsymbol{s}$, a player i, and assume i deviates to a new set of resources $s'_i \in S_i$. For convenience, for every $e \in E$, denote $n_e = n_e(\boldsymbol{s})$ and $n'_e = n_e(s'_i, \boldsymbol{s}_{-i})$. Then notice that, given that all other players remain fixed at $\boldsymbol{s}_{-i}$, for all resources $e \in s'_i \cap s_i$ that player i keeps occupying, and for all resources $e \in E \setminus (s_i \cup s'_i)$ that player i does not play in either of her strategies, the congestion remains unchanged, i.e. $n'_e = n_e$. On the other hand, for all "new" resources $e \in s'_i \setminus s_i$ we have that $n'_e = n_e + 1$, and for all "abandoned" resources $e \in s_i \setminus s'_i$ we have $n'_e = n_e - 1$.

Putting all these together, we get that

$$\Phi(s'_i, \boldsymbol{s}_{-i}) - \Phi(\boldsymbol{s}) = \sum_{e\in E} \sum_{j=1}^{n'_e} c_e(j) - \sum_{e\in E} \sum_{j=1}^{n_e} c_e(j)$$

$$
\begin{aligned}
&= \sum_{e\in E}\left(\sum_{j=1}^{n'_e} c_e(j) - \sum_{j=1}^{n_e} c_e(j)\right) \\
&= \sum_{e\in s'_i\setminus s_i}\left(\sum_{j=1}^{n_e+1} c_e(j) - \sum_{j=1}^{n_e} c_e(j)\right) + \sum_{e\in s_i\setminus s'_i}\left(\sum_{j=1}^{n_e-1} c_e(j) - \sum_{j=1}^{n_e} c_e(j)\right) \\
&+ \sum_{e\in s'_i\cap s_i}\left(\sum_{j=1}^{n_e} c_e(j) - \sum_{j=1}^{n_e} c_e(j)\right) + \sum_{e\in E\setminus(s_i\cup s'_i)}\left(\sum_{j=1}^{n_e} c_e(j) - \sum_{j=1}^{n_e} c_e(j)\right) \\
&= \sum_{e\in s'_i\setminus s_i} c_e(n_e+1) + \sum_{e\in s_i\setminus s'_i} (-c_e(n_e)) + 0 + 0 \\
&= \sum_{e\in s'_i\setminus s_i} c_e(n_e+1) - \sum_{e\in s_i\setminus s'_i} c_e(n_e),
\end{aligned}
$$

while

$$
\begin{aligned}
C_i(s'_i, \boldsymbol{s}_{-i}) - C_i(\boldsymbol{s}) &= \sum_{e\in s'_i} c_e(n'_e) - \sum_{e\in s_i} c_e(n_e) \\
&= \left(\sum_{e\in s'_i\setminus s_i} c_e(n'_e) + \sum_{e\in s'_i\cap s_i} c_e(n'_e)\right) - \left(\sum_{e\in s_i\setminus s'_i} c_e(n_e) + \sum_{e\in s_i\cap s'_i} c_e(n_e)\right) \\
&= \sum_{e\in s'_i\setminus s_i} c_e(n_e+1) + \sum_{e\in s'_i\cap s_i} c_e(n_e) - \sum_{e\in s_i\setminus s'_i} c_e(n_e) - \sum_{e\in s_i\cap s'_i} c_e(n_e) \\
&= \sum_{e\in s'_i\setminus s_i} c_e(n_e+1) - \sum_{e\in s_i\setminus s'_i} c_e(n_e),
\end{aligned}
$$

thus indeed satisfying definition (5). □

2.3 Price of Anarchy and Stability

Fix a potential game $\mathcal{G}$. Let $\mathrm{PNE}(\mathcal{G})$ denote the set of all pure Nash equilibria of $\mathcal{G}$ (which is nonempty, due to Theorem 2.1) and $\boldsymbol{s}^* \in \operatorname{argmin}_{\boldsymbol{s}\in\boldsymbol{S}} C(\boldsymbol{s})$ a *(socially) optimum* outcome. Then, we can define the following two notions, known as the *Price of Anarch (PoA)* and the *Price of Stability (PoS)*. They correspond to the (multiplicative) gap of the social cost between the worst and best, respectively, equilibria to the social optimum:

$$
\mathrm{PoA}(\mathcal{G}) = \frac{\max_{\boldsymbol{s}\in\mathrm{PNE}(\mathcal{G})} C(\boldsymbol{s})}{C(\boldsymbol{s}^*)} \qquad \mathrm{PoS}(\mathcal{G}) = \frac{\min_{\boldsymbol{s}\in\mathrm{PNE}(\mathcal{G})} C(\boldsymbol{s})}{C(\boldsymbol{s}^*)}.
$$

To handle the corner cases in the above expression, we take $\frac{0}{0} = 1$.

Obviously, for any potential game $\mathcal{G}$ it is: $1 \le \mathrm{PoS}(\mathcal{G}) \le \mathrm{PoA}(\mathcal{G})$.

The PoA and PoS notions can be extended, in a worst-case manner, to *classes* of games Γ via

$$
\mathrm{PoA}(\Gamma) = \sup_{\mathcal{G}\in\Gamma} \mathrm{PoA}(\mathcal{G}) \quad \text{and} \quad \mathrm{PoS}(\Gamma) = \sup_{\mathcal{G}\in\Gamma} \mathrm{PoS}(\mathcal{G})
$$

For example, going back to Example 2.2, let $\mathcal{G}_\varepsilon$ denote the Pigou network of Figure 1. Then, given that $\mathcal{G}_\varepsilon$ has a unique PNE, namely $(Down, Down)$, with a social cost of $2 + 2 = 4$, while the social optimum is $2 + \varepsilon + 1 = 3 + \varepsilon$ (given, e.g., by $(Up, Down)$), we have that $\mathrm{PoA}(\mathcal{G}_\varepsilon) = \mathrm{PoS}(\mathcal{G}_\varepsilon) = \frac{4}{3+\varepsilon}$. If we consider the class of all these Pigou networks $\Gamma = \{\mathcal{G}_\varepsilon\}_{\varepsilon\in(0,1)}$ we have $\mathrm{PoA}(\Gamma) = \mathrm{PoS}(\Gamma) = \frac{4}{3}$.

Theorem 2.3 (Christodoulou and Koutsoupias [2005]). *The Price of Anarchy of linear congestion games is at most* $5/2$.

Proof. Fix an arbitrary linear congestion game and let $\boldsymbol{s}^*$ be a strategy profile minimizing its social cost. We need to show that for any PNE $\boldsymbol{s}$:

$$C(\boldsymbol{s}) \leq \frac{5}{2} \cdot C(\boldsymbol{s}^*). \tag{9}$$

To simplify notation, for the remainder of the proof we will denote $n_e = n_e(\boldsymbol{s})$, $n_e^* = n_e(\boldsymbol{s}^*)$ for all resources $e \in E$. Furthermore, we know that the cost functions are of the form $c_e(x) = a_e x + b_e$, where $a_e, b_e \in \mathbb{R}_{\geq 0}$. Then, we can express the social costs of the equilibrium and optimum profiles as:

$$C(s) = \sum_{e\in E} n_e c_e(n_e) = \sum_{e\in E} n_e \left(a_e n_e + b_e\right) = \sum_{e\in E} \left(a_e n_e^2 + b_e n_e\right) \tag{10}$$

$$C(\boldsymbol{s}^*) = \sum_{e\in E} \left(a_e (n_e^*)^2 + b_e n_e^*\right). \tag{11}$$

Since $\boldsymbol{s}$ is a PNE, it must be (see (7)) that, for every player i,

$$\sum_{e\in s_i} c_e(n_e) \leq \sum_{e\in s_i^*\cap s_i} c_e(n_e) + \sum_{e\in s_i^*\setminus s_i} c_e(n_e+1) \leq \sum_{e\in s_i^*} c_e(n_e+1).$$

Summing over $i \in N$ we get:

$$\begin{aligned} C(\boldsymbol{s}) &= \sum_{i\in N}\sum_{e\in s_i} c_e(n_e) \\ &\leq \sum_{i\in N}\sum_{e\in s_i^*} c_e(n_e+1) \\ &= \sum_{e\in E} n_e^* c_e(n_e+1) \\ &= \sum_{e\in E} n_e^* \left(a_e(n_e+1) + b_e\right) \\ &= \sum_{e\in E} \left[a_e n_e^*(n_e+1) + b_e n_e^*\right]. \end{aligned} \tag{12}$$

Now, we will need the following algebraic lemma[1]:

Lemma 2.1. *For all* $y, z \in \mathbb{N}$:

$$y(z+1) \leq \frac{5}{3}y^2 + \frac{1}{3}z^2.$$

Applying Lemma 2.1 to (12) for $y \leftarrow n_e^*$ and $z \leftarrow n_e$ we have that:

$$\begin{aligned} C(\boldsymbol{s}) &\leq \sum_{e\in E}\left[a_e\left(\frac{5}{3}(n_e^*)^2 + \frac{1}{3}n_e^2\right) + b_e n_e^*\right] \\ &= \sum_{e\in E}\left[\frac{5}{3}a_e(n_e^*)^2 + b_e n_e^*\right] + \sum_{e\in E}\frac{1}{3}a_e n_e^2 \\ &\leq \frac{5}{3}\sum_{e\in E}\left[a_e(n_e^*)^2 + b_e n_e^*\right] + \frac{1}{3}\sum_{e\in E}\left[a_e n_e^2 + b_e n_e\right] \\ &= \frac{5}{3}C(\boldsymbol{s}^*) + \frac{1}{3}C(\boldsymbol{s}), \end{aligned}$$

[1] Its proof will be an exercise in your next assignment.

the last equality holding due to (10) and (11). Moving the equilibrium social cost $C(\boldsymbol{s})$ on the left-hand side of the above inequality, we finally get

$$C(\boldsymbol{s}) - \frac{1}{3}C(\boldsymbol{s}) \leq \frac{5}{3}C(\boldsymbol{s}^*),$$

which is equivalent to the desired (9). □

Theorem 2.4 (Christodoulou and Koutsoupias [2005])**.** *The Price of Anarchy of linear congestion games is at least* $5/2$*.*

Proof. Consider a congestion game with three players $N = \{1, 2, 3\}$ and six resources $E = \{e_1, e_2, e_3, \bar{e}_1, \bar{e}_2, \bar{e}_3\}$. Each player $i \in N$ has two strategies $S_i = \{s_i^*, s_i\}$, namely

$$s_i^* = \{e_i, \bar{e}_i\} \quad \text{and} \quad s_i = \{e_{i+1}, \bar{e}_{i+1}, \bar{e}_{i+2}\},$$

where $e_j = e_{j-3}$ and $\bar{e}_j = \bar{e}_{j-3}$ for $j > 3$. All resources have the same cost function $c_e(x) = x$, for all $e \in E$.

We will show that the strategy profile $\boldsymbol{s} = (s_1, s_2, s_3)$ is a PNE, having social cost $C(\boldsymbol{s}) = 15$, while the outcome $\boldsymbol{s}^* = (s_1^*, s_2^*, s_3^*)$ has social cost $C(\boldsymbol{s}^*) = 6$. This would be enough to establish the theorem, since the PoA of the game would be at least $C(\boldsymbol{s})/C(\boldsymbol{s}^*) = 15/6 = 5/2$.

It is not hard to see that at outcome $\boldsymbol{s}$, resources e_1, e_2, e_3 are used by 1 player, while resources $\bar{e}_1, \bar{e}_2, \bar{e}_3$ are used by 2 players. This results in a social cost of $C(\boldsymbol{s}) = 3 \cdot 1 \cdot c(1) + 3 \cdot 2 \cdot c(2) = 15$. On the other hand, on outcome $\boldsymbol{s}^*$ all resources are used by exactly 1 player, thus $C(\boldsymbol{s}^*) = 6 \cdot 1 \cdot c(1) = 6$.

Finally, to verify that $\boldsymbol{s}$ is indeed an equilibrium, observe that player 1 on $\boldsymbol{s}$ is playing on resources $e_2, \bar{e}_2, \bar{e}_3$, having loads of 1, 2 and 2, respectively. If she would (unilaterally) deviate to strategy s_1^*, then she would play on resources $e_1, \bar{e}_1$ which would now have a load of 2 and 3, respectively. This shows that $C_1(\boldsymbol{s}) = 1 + 2 + 2 = 5 \leq 2 + 3 = C_1(s_1^*, \boldsymbol{s}_{-1})$. The analysis for players $i = 2, 3$ is analogous. □

Combining Theorem 2.3 and Theorem 2.4, we deduce that the PoA of the class of linear congestion games is *exactly* $5/2$.

Theorem 2.5. *The Price of Stability of polynomial congestion games of degree* d *is at most* $d + 1$*.*

Proof. We will make use of the following (more general) tool:

Lemma 2.2 (Potential Method)**.** *Assume that* Φ *is a potential function for (cost-minimization) game* $\mathcal{G}$*. If there exist constants* $\alpha, \beta > 0$ *such that, for any strategy profile* $\boldsymbol{s}$*,*

$$\alpha \cdot C(\boldsymbol{s}) \leq \Phi(\boldsymbol{s}) \leq \beta \cdot C(\boldsymbol{s}), \tag{13}$$

then PoS$(\mathcal{G}) \leq \frac{\beta}{\alpha}$*.*

Proof of Lemma 2.2. Since the Price of Stability is defined as the cost of the *best* pure Nash equilibrium over the optimum social cost, it is enough if we show that there exists a pure equilibrium $\tilde{\boldsymbol{s}}$ such that

$$C(\tilde{\boldsymbol{s}}) \leq \frac{\beta}{\alpha} C(\boldsymbol{s}^*),$$

where $\boldsymbol{s}^* \in \operatorname{argmin}_{\boldsymbol{s}} C(\boldsymbol{s})$.

We choose $\tilde{\boldsymbol{s}} \in \operatorname{argmin}_{\boldsymbol{s}} \Phi(\boldsymbol{s})$ to be a minimizer of the potential. We know (from the proof of Theorem 2.1) that $\tilde{\boldsymbol{s}}$ is indeed an equilibrium. Furthermore, using (13) we get that

$$\alpha C(\tilde{\boldsymbol{s}}) \leq \Phi(\tilde{\boldsymbol{s}}) \leq \Phi(\boldsymbol{s}^*) \leq \beta C(\boldsymbol{s}^*),$$

where the middle inequality holds because $\tilde{\boldsymbol{s}}$ is a potential minimizer. ■

From Theorem 2.2 we know that Rosenthal's function Φ from (8) is a potential for congestion games. Furthermore, one can show[2] that polynomial congestion games of degree d satisfy condition (13) with $\alpha = \frac{1}{d+1}$ and $\beta = 1$. Therefore, using Lemma 2.2 we immediately get that indeed the PoS is upper-bounded by $\beta/\alpha = 1/(1/(d+1)) = d+1$. □

2.4 Computation of Equilibria

Throughout this section we use the following notation for some critical parameters of our congestion games: $n = |N|$, $m = |E|$, $k_i = |S_i|$ for all players i, and $k = \max_{i\in N} k_i$. Furthermore, we will assume[3] that resource costs take integer values, i.e., $c_e(j) \in \mathbb{N}$ for all $e \in E$, $j \in [n]$.

2.4.1 Input Representation

Before we formally study the complexity of computing equilibria in congestion games, we need to fix some important aspects regarding the input size for our algorithms, and in particular how congestion game instances are going to be represented. The *size* of the description of a congestion game $\mathcal{G}$ is determined by:

- the number of players $n = |N|$,
- the number of resources $m = |E|$,
- the number of (binary) bits needed to represent the cost functions. More specifically, these can be given either
  - *explicitly*, in which case the list of costs $\{c_e(j)\}_{e\in E, j\in[n]}$ is given in the input, and thus they require a total space of $O(mn \log c_{\max})$ bits, where $c_{\max} \coloneqq \max_e c(n)$; or
  - *implicitly*, for example like in the case of (degree d) polynomial congestion games, where we can compute the costs $c_e(x) = \sum_{j=0}^{d} a_{e,j} x^j$ of resource e just by knowing its coefficients $\{a_{e,j}\}_{j=0,1,\dots,d}$. This takes up a space of $O(md \log a_{\max})$ in total, where $a_{\max} \coloneqq \max_{e,j} a_{e,j}$.
- the strategy sets $\{S_i\}_{i\in N} \subseteq 2^E$ of the players. These can also be given either
  - *explicitly*, in which case each S_i is described by listing all its elements (which are sets of resources), thus resulting in a total size of $O(nkm)$, where $k = \max_i |S_i|$; or
  - *implicitly*, for example like in network congestion games. In this case, the strategies of each player i (which can be exponentially many with respect to m) are not explicitly listed in the input, but just an origin-destination node pair (o_i, d_i). This description has a size of $O(n)$. It is important to emphasize that, for such implicit representations that the player has not explicit access to all her strategy set, she must still be able to check (in polynomial time) whether a given strategy profile $\boldsymbol{s}$ is a PNE, and if not, find an improving deviation $s_i' \in S_i$ that reduces her cost. For example, in network congestion games this can be done via a shortest-path computation.

To avoid some of these technicalities, it is common to consider inputs where the values of $c_{\max}$, $a_{\max}$ are at most exponential on n and m, and k is polynomial. Then, the size of the input boils down to being polynomial on n and m, and efficient algorithms are those who run in time $O(\mathrm{poly}(n, m))$.

From the above analysis, one can immediately come up with a first (naive) algorithm for computing a PNE of a congestion game: exhaustively search all outcomes $\boldsymbol{s} \in \boldsymbol{S}$, each time checking if $\boldsymbol{s}$ is a PNE.

[2]We leave this as an exercise for your next assignment.

[3]This is essentially without loss for our computational results, since all rational costs can be scaled appropriately to integers (with just a polynomial effect on the complexity).

Of course, since in general there can be exponentially[4] many outcomes, this is not an efficient method for computing equilibria.

2.4.2 Better-Response Dynamics

The proof of Theorem 2.1 readily suggests another natural method for computing equilibria in potential games: if the current outcome $\boldsymbol{s}$ is not a PNE, allow a player to unilaterally switch to a beneficial deviation. This is known as *better-response dynamics (BRD)*:

Algorithm 1 Better-Response Dynamics (BRD)

Input: Game $\mathcal{G} = (N, \{S_i\}, \{C_i\})$; profile $\boldsymbol{s} \in S$
Output: A pure Nash equilibrium $\boldsymbol{s}$ of $\mathcal{G}$
1: **while** $\boldsymbol{s}$ is not a PNE **do**
2:     Choose $i \in N$, $s_i' \in S_i$ such that: $C_i(s_i', \boldsymbol{s}_{-i}) < C_i(\boldsymbol{s})$
3:     $\boldsymbol{s} \leftarrow (s_i', \boldsymbol{s}_{-i})$
4: **end while**
5: **return** $\boldsymbol{s}$

BRD allows for great flexibility in the choice of the improving move in Step 2 of Algorithm 1. One can restrict this further in order to get the following two interesting variants of BRD:

- *Best-response dynamics:* the *best* deviation of player i is chosen. Formally, Line 2 of Algorithm 1 is complemented by condition:
$$s_i' \in \operatorname*{argmin}_{\tilde{s} \in S_i} C_i(\tilde{s}, \boldsymbol{s}_{-i}).$$
- *Maximum-gain best-response dynamics:* the player with the best (relative) deviation is chosen. Formally, Line 2 of Algorithm 1 is complemented by condition:
$$(i, s_i') \in \operatorname*{argmin}_{(j, \tilde{s})} \frac{C_j(\tilde{s}, \boldsymbol{s}_{-j})}{C_j(\boldsymbol{s})}. \tag{14}$$

Theorem 2.6. *Better-response dynamics find a pure Nash equilibrium of a congestion game after at most $nmc_{\max}$ iterations.*

Proof. From the proof of Theorem 2.2 we know that Rosenthal's function Φ (see (8)) is a potential for any congestion game. Thus, by Definition 2.1, it must be the case that after every iteration of the while-loop of Algorithm 1, the value of $\Phi(\boldsymbol{s})$ *strictly* decreases. Given our assumption of integer resource costs, this means that BRD have to terminate after at most $\Phi_{\max} - \Phi_{\min}$ iterations, where

$$\Phi_{\max} := \max_{\boldsymbol{s}} \Phi(\boldsymbol{s}) = \max_{\boldsymbol{s}} \sum_{e} \sum_{j=1}^{n_e(\boldsymbol{s})} c_e(j) \leq \sum_{e} nc_{\max} = mnc_{\max} \tag{15}$$
$$\Phi_{\min} := \min_{\boldsymbol{s}} \Phi(\boldsymbol{s}) \geq 0,$$

which gives us the desired bound on the number of iterations. □

Notice that the bound in Theorem 2.6 is, in general, still exponential in the size of the input. However, it becomes polynomial when the values of the cost functions are polynomially bounded. In other words, BRD have *pseudopolynomial* running time.

[4] We can only guarantee that $|S| = \prod_{i=1}^{n} k_i \leq k^n \leq (2^m)^n = 2^{nm}$.

2.4.3 Singleton Games

Theorem 2.7 (Ieong et al. [2005])**.** *In singleton congestion games, a PNE can be computed in polynomial time.*

Proof. Fix a singleton congestion game $\mathcal{G} = (N, E, \{S_i\}, \{c_e\})$. Our goal is to construct an "equivalent" game $\tilde{\mathcal{G}} = (N, E, \{S_i\}, \{\tilde{c}_e\})$ with $\tilde{c}_{\max} = \text{poly}(n, m)$. Then, we can deploy Theorem 2.6 to immediately get convergence of BRD in polynomially many steps, concluding our proof.

Let $K = \{c_e(j) \mid e \in E, j \in [n]\}$ be the set of all possible resource-cost values of $\mathcal{G}$. Notice that $|K| \leq nm$. We define modified cost functions $\tilde{c}_e$, for all resources $e \in E$, via

$$\tilde{c}_e(j) = k \quad \iff \quad |\{\kappa \in K \mid \kappa < c_e(j)\}| = k - 1. \tag{16}$$

In other words, $\tilde{c}_e(j)$ gives the position of the value $c_e(j)$ within K (when taken ordered from smallest to largest). By this, we can immediately deduce that indeed $\tilde{c}_e(j) \leq nm = \text{poly}(n, m)$ for all resources e.

The equivalence of $\mathcal{G}$ and $\tilde{\mathcal{G}}$ can now be taken to mean that the two games preserve the better-response incentives of the players. Formally, we need to show that for all outcomes $\boldsymbol{s}$, every player i and any deviation s_i':

$$C_i(s_i', \boldsymbol{s}_{-i}) < C_i(\boldsymbol{s}) \quad \iff \quad \tilde{C}_i(s_i', \boldsymbol{s}_{-i}) < \tilde{C}_i(\boldsymbol{s}).$$

Making use of the fact that our games are singleton, and denoting with e, e' the (single) resource in strategies s_i and s_i', respectively, this condition is equivalently written as

$$c_{e'}(n_{e'} + 1) < c_e(n_e) \quad \iff \quad \tilde{c}_{e'}(n_{e'} + 1) < \tilde{c}_e(n_e), \tag{17}$$

where for simplicity we are using the shortcut notation $n_e = n_e(\boldsymbol{s})$, $n_{e'} = n_{e'}(\boldsymbol{s})$. Now it is not hard to establish the validity of (17), directly by observing that the definition of the modified costs $\tilde{c}_e$ in (16) preserves the relative order of the cost values. □

2.4.4 Approximate Equilibria and Symmetric Games

Definition 2.2 (Bounded jump)**.** Fix a congestion game $\mathcal{G}$ and let $\alpha \geq 1$. We say that $\mathcal{G}$ has the α-bounded jump property if, for all resources e and any positive integer j,

$$c_e(j + 1) \leq \alpha \cdot c_e(j).$$

Definition 2.3 (Approximate pure Nash equilibria)**.** Fix a (finite, cost-minimization) game $\mathcal{G}$ and let $\alpha \geq 1$. An outcome $\boldsymbol{s}$ is an *α-approximate pure Nash equilibrium (α-PNE)* of $\mathcal{G}$ if, for every player i and every deviation $s_i' \in S_i$,

$$C_i(\boldsymbol{s}) \leq \alpha \cdot C_i(s_i', \boldsymbol{s}_{-i}).$$

For the special case of $\alpha = 1$ one recovers the standard notion of an (exact) PNE (see (2)). Notice that, every α-PNE is also an α'-PNE, for any $\alpha' \geq \alpha \geq 1$.

By Definition 2.3, if a profile $\boldsymbol{s}$ is *not* an α-PNE (for some $\alpha \geq 1$), there must exist a player i and a deviation $s_i' \in S_i$ such that

$$\alpha C_i(s_i', \boldsymbol{s}_{-i}) < C_i(\boldsymbol{s})$$

Such an $\boldsymbol{s} \longrightarrow (s_i', \boldsymbol{s}_{-i})$ move will be called *α-improving*. Thus, a 1-improving move is exactly a better-response deviation in the spirit of Line 2 of Algorithm 1. Furthermore, an α-improving move is α'-improving as well, for all $1 \leq \alpha' \leq \alpha$. In particular, notice that if a profile $\boldsymbol{s}$ is not an α-PNE and $\boldsymbol{s} \longrightarrow (s_i', \boldsymbol{s}_{-i})$ is a move chosen by any maximum-gain best-response dynamics (see (14)), then $\boldsymbol{s} \longrightarrow (s_i', \boldsymbol{s}_{-i})$ must be (at least) α-improving.

Theorem 2.8 (Chien and Sinclair [2011]). *In symmetric congestion games with the α-bounded jump property, an $(1+\varepsilon)$-PNE can be computed in time polynomial on $\frac{1}{\varepsilon}$, α, and the size of the input.*

Proof. Fix a symmetric congestion game $\mathcal{G}$, having the α-bounded jump property, and some $\varepsilon > 0$. We will show that maximum-gain best-response dynamics (see (14)), started from an arbitrary strategy profile $\boldsymbol{s}^0$, always find a $(1+\varepsilon)$-approximate PNE after at most

$$\left(1+\frac{1}{\varepsilon}\right)\alpha n \ln\left(nmc_{\max}\right)$$

iterations.

We will need the following lemmas:

Lemma 2.3. *For any outcome* $\boldsymbol{s}$,

$$\max_{i\in N} C_i(\boldsymbol{s}) \geq \frac{1}{n}\Phi(\boldsymbol{s}).$$

Proof. For every profile $\boldsymbol{s}$ it is

$$\Phi(\boldsymbol{s}) = \sum_{e\in E}\sum_{j=1}^{n_e(\boldsymbol{s})} c_e(j) \leq \sum_{e\in E} n_e(\boldsymbol{s})c_e(n_e(\boldsymbol{s})) = C(\boldsymbol{s}) = \sum_{i=1}^{n} C_i(\boldsymbol{s})$$

and thus

$$\max_{i\in N} C_i(\boldsymbol{s}) \geq \frac{1}{n}\sum_{i=1}^{n} C_i(\boldsymbol{s}) \geq \frac{1}{n}\Phi(\boldsymbol{s}).$$

■

Lemma 2.4. *If player i and a move $\boldsymbol{s} \longrightarrow (s_i', \boldsymbol{s}_{-i})$ are chosen at some step of the maximum-gain best-response dynamics, then*

$$C_i(\boldsymbol{s}) \geq \frac{1}{\alpha}\max_{j\in N} C_j(\boldsymbol{s}).$$

Proof. We need to show that, for all players j:

$$C_i(\boldsymbol{s}) \geq \frac{1}{\alpha}C_j(\boldsymbol{s}). \tag{18}$$

Fix an arbitrary player j. First notice that, since $\mathcal{G}$ is symmetric, $(s_i', \boldsymbol{s}_{-j})$ is a valid strategy profile as well. Comparing the strategy profiles $(s_i', \boldsymbol{s}_{-i})$, $(s_i', \boldsymbol{s}_{-j})$ we observe that all players except i and j use the same strategies in both of them and, furthermore, s_i' is played in both. Therefore, the two profiles have at least $n-1$ strategies in common and so, the number of players that use any resource e can differ at most by 1. Formally, for all $e \in E$,

$$\left|n_e(s_i', \boldsymbol{s}_{-i}) - n_e(s_i', \boldsymbol{s}_{-j})\right| \leq 1.$$

By the α-bounded jump property, this gives

$$c_e\left(n_e(s_i', \boldsymbol{s}_{-j})\right) \leq c_e\left(n_e(s_i', \boldsymbol{s}_{-i}) + 1\right) \leq \alpha c_e\left(n_e(s_i', \boldsymbol{s}_{-i})\right).$$

Summing the left- and right-hand sides of this inequality over all $e \in s_i'$, we can deduce that

$$C_j(s_i', \boldsymbol{s}_{-j}) \leq \alpha C_i(s_i', \boldsymbol{s}_{-i}). \tag{19}$$

Next notice that, since $\boldsymbol{s} \longrightarrow (s'_i, \boldsymbol{s}_{-i})$ is a maximum-gain move, from (14) we know that it must be

$$\frac{C_i(s'_i, \boldsymbol{s}_{-i})}{C_i(\boldsymbol{s})} \leq \frac{C_j(s'_i, \boldsymbol{s}_{-j})}{C_j(\boldsymbol{s})}.$$

Using this, we can finally bound:

$$C_i(\boldsymbol{s}) \geq \frac{C_i(s'_i, \boldsymbol{s}_{-i})}{C_j(s'_i, \boldsymbol{s}_{-j})} C_j(\boldsymbol{s}) \overset{(19)}{\geq} \frac{1}{\alpha} C_j(\boldsymbol{s}),$$

proving (18). ■

Returning now to the proof of Theorem 2.8, we can bound the decrease in Rosenthal's potential at every step that our dynamics make an $(1+\varepsilon)$-improving move $\boldsymbol{s} \longrightarrow (s'_i, \boldsymbol{s}_{-i})$, by

$$\begin{aligned}
\Phi(\boldsymbol{s}) - \Phi(s'_i, \boldsymbol{s}_{-i}) &= C_i(\boldsymbol{s}) - C_i(s'_i, \boldsymbol{s}_{-i}) \\
&\geq C_i(\boldsymbol{s}) - \frac{C_i(\boldsymbol{s})}{1+\varepsilon}, && (1+\varepsilon)\text{-improving move,} \\
&= \frac{\varepsilon}{1+\varepsilon} C_i(\boldsymbol{s}) \\
&\geq \frac{\varepsilon}{\alpha(1+\varepsilon)} \max_{j \in N} C_j(\boldsymbol{s}), && \text{from Lemma 2.4,} \\
&\geq \frac{\varepsilon}{n\alpha(1+\varepsilon)} \Phi(\boldsymbol{s}), && \text{from Lemma 2.3,}
\end{aligned}$$

and so

$$\Phi(s'_i, \boldsymbol{s}_{-i}) \leq \left(1 - \frac{\varepsilon}{n\alpha(1+\varepsilon)}\right) \Phi(\boldsymbol{s}). \tag{20}$$

Since we are performing maximum-gain dynamics (14), every move $\boldsymbol{s} \longrightarrow (s'_i, \boldsymbol{s}_{-i})$ from a state $\boldsymbol{s}$ which is *not* yet an $(1+\varepsilon)$-PNE, has to be an $(1+\varepsilon)$-improving move. Therefore, if $\boldsymbol{s}^\ell$ denotes the outcome that we have reached after ℓ iterations for which our dynamics have not yet found an $(1+\varepsilon)$-PNE, due to (20) the potential has a value of at most

$$\Phi(\boldsymbol{s}^\ell) \leq \left(1 - \frac{\varepsilon}{n\alpha(1+\varepsilon)}\right)^\ell \Phi(\boldsymbol{s}^0).$$

Taking logarithms, we get that

$$\ell \cdot \ln\left(1 - \frac{\varepsilon}{n\alpha(1+\varepsilon)}\right) \geq \ln \frac{\Phi(\boldsymbol{s}^\ell)}{\Phi(\boldsymbol{s}^0)},$$

and making use of the inequality $\ln(1-x) \leq -x$ (holding for all $x \geq 0$), we can finally derive that

$$\ell \leq \frac{\alpha n(1+\varepsilon)}{\varepsilon} \ln \frac{\Phi(\boldsymbol{s}^0)}{\Phi(\boldsymbol{s}^m)} \leq n\alpha\left(1 + \frac{1}{\varepsilon}\right) \ln \Phi_{\max} \leq n\alpha\left(1 + \frac{1}{\varepsilon}\right) \ln(mnc_{\max}),$$

where in the last step we used the potential upper bound from (15). □

2.4.5 Network Symmetric Games

Theorem 2.9 (Fabrikant, Papadimitriou, and Talwar [2004]). *In symmetric network congestion games, a PNE can be computed in polynomial time.*

Proof. Fix a network congestion game $\mathcal{G}$ on a directed graph $G = (V, E)$, with resource/edge cost functions $\{c_e\}_{e\in E}$, where the strategies of all players are $o \to d$ paths on G (where $o, d \in V$). We will show that a minimizer $\boldsymbol{s}^* \in \operatorname{argmin}_{\boldsymbol{s}\in S} \Phi(\boldsymbol{s})$ of Rosenthal's potential can be computed in polynomial time (in the description of $\mathcal{G}$), via a reduction to the minimum-cost flow problem.

In general, an instance of Min-Cost Flow consists of a directed graph $\tilde{G} = (\tilde{V}, \tilde{E})$ with edge capacities $u : \tilde{E} \longrightarrow \mathbb{R}_{\geq 0}$ and costs $\kappa : \tilde{E} \longrightarrow \mathbb{R}$, and node demands $b : \tilde{V} \longrightarrow \mathbb{R}$ where $\sum_{v\in\tilde{V}} b(v) = 0$. The objective is to find a feasible flow $f : \tilde{E} \longrightarrow \mathbb{R}_{\geq 0}$ on $\tilde{G}$ that induces the minimum total cost. Formally, Min-Cost Flow asks for the solution to the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \sum_{e\in\tilde{E}} f(e)\kappa(e) && \\ \text{such that} \quad & \sum_{e\in\delta^+(v)} f(e) - \sum_{e\in\delta^-(v)} f(e) = b(v) && \forall v \in \tilde{V} \\ & 0 \leq f(e) \leq u(e) && \forall e \in \tilde{E}, \end{aligned} \tag{21}$$

where $\delta^+(v)$, $\delta^-(v)$ denote the set of edges leaving from and arriving to vertex v, respectively. It is known (see, e.g., Cook et al. [1998, Ch. 4] or Wolsey [2020, Sec. 3.3]) that if demands b and capacities u are integers, then Problem (21) has an optimal *integral* solution (i.e., with $f(e) \in \mathbb{N}$ for all e) and, furthermore, such an optimal solution can be computed in polynomial time.

For the reduction, we construct the following Min-Cost Flow instance from the original congestion game $\mathcal{G}$: we take the same set of vertices $\tilde{V} = V$ and for every edge $e \in E$ we introduce $n = |N|$ parallel edges $\tilde{e}_1, \dots, \tilde{e}_n$ (with the same end-points as e) in $\tilde{E}$, with unit capacities $u(\tilde{e}_j) = 1$ and costs $\kappa(\tilde{e}_j) = c_e(j)$, $j = 1, 2, \dots, n$. We set the demands to $b(o) = n$, $b(d) = -n$ and $b(v) = 0$ for all $v \in \tilde{V} \setminus \{o, d\}$; that is, we essentially allow only $o \to d$ flows of size equal to the number of players. Under this construction (which can be carried out in polynomial time on the size of $\mathcal{G}$), we can recover a minimizer $\boldsymbol{s}^*$ of Rosenthal's potential from an *integral* minimum-cost flow f^* in the following way: if parallel edges $\tilde{e}_1, \tilde{e}_2, \dots, \tilde{e}_j$ are used by f^* while $\tilde{e}_{j+1}, \dots, \tilde{e}_n$ are not (i.e., j is such that $f^*(\tilde{e}_\ell) = 1$ for $\ell = 1, \dots, j$ and $f^*(\tilde{e}_\ell) = 0$ for $\ell = j + 1, \dots, n$) then $n_e(\boldsymbol{s}^*) = j$. As a matter of fact, it is not hard to see that the objective value of the minimum-cost flow f^* is equal to the minimum potential value:

$$\sum_{\tilde{e}\in\tilde{E}} f^*(\tilde{e})\kappa(\tilde{e}) = \sum_{e\in E}\sum_{\ell=1}^{n} f^*(\tilde{e}_\ell)\kappa(\tilde{e}_\ell) = \sum_{e\in E}\sum_{\ell=1}^{n_e(\boldsymbol{s}^*)} c_e(\ell) = \Phi(\boldsymbol{s}^*).$$

□

3 Computation of Pure Nash Equilibria

3.1 Local Search and the Class PLS

3.1.1 Better-Response Graphs

Consider a congestion game $\mathcal{G} = (N, R, \{S_i\}_{i \in N}, \{c_r\}_{r \in R})$ and let Φ denote its (Rosenthal) potential (see Theorem 2.2). The *better-response graph* of $\mathcal{G}$ is the graph consisting of all edges pointing from a strategy profile to a single-player beneficial deviation. Formally, it is the directed graph $G = (V, E)$ with

$$V = S \qquad \text{and} \qquad E = \left\{ (\boldsymbol{s}, \boldsymbol{s}') \;\middle|\; \boldsymbol{s}' = (s_i', \boldsymbol{s}_{-i}), s_i' \in S_i, i \in N \;\text{ and }\; C_i(\boldsymbol{s}') < C_i(\boldsymbol{s}) \right\}.$$

Observe that, along every path of G the potential of the nodes is strictly decreasing (see Definition 2.1). Therefore, G is *acyclic* and has at least one *sink* (that is, a node with out-degree 0). It is easy to see that every such sink $\tilde{\boldsymbol{s}}$ satisfies

$$\Phi(\tilde{\boldsymbol{s}}) \leq \Phi(s_i', \tilde{\boldsymbol{s}}_{-i}) \qquad \text{for all } \; i \in N, s_i' \in S_i \tag{22}$$

and that the sinks of G correspond *exactly* to the PNE of $\mathcal{G}$. As a matter of fact, every execution of BRD (Algorithm 1) can be interpreted as walking along a path of G until a sink is reached.

3.1.2 Maximum Cuts

Fix an undirected graph $G = (V, E)$ with edge-weights $w(e) > 0$. We will use $w(E') = \sum_{e \in E'} w(e)$ to denote the total weight of a subset of edges $E' \subseteq E$ and

$$E(A, B) = \{\{u, v\} \in E \mid u \in A, v \in B\}$$

for the set of edges between nodes in $A, B \subseteq V$.

A *cut* of G is a partition $(K, V \setminus K)$ of the set of nodes V. It is standard to denote such a cut simply by one of its partitions $K \subseteq V$ (since the other $V \setminus K$ is fully determined by K). The *weight of a cut* K is defined as the total weight of the edges crossing it, i.e.

$$W(K) \coloneqq w(E(K, V \setminus K)).$$

Finding a cut of maximum weight $\max_{K \subseteq V} W(K)$, which is known as the Max-Cut problem, is NP-hard even for graphs with unit weights[5] (see, e.g., Garey et al. [1976] or [Papadimitriou, 1994, Theorem 9.5]). Here we are interested in a relaxed version of the problem, where the goal is to find a *local* optimum, that is, a cut whose weight cannot be further improved by moving a single node at the opposite side. Formally, Local-Max-Cut asks for a cut $K \subseteq V$ such that

$$W(K \setminus \{v\}) \leq W(K), \qquad \text{for all } \; v \in K, \tag{23}$$

$$W(K \cup \{v\}) \leq W(K), \qquad \text{for all } \; v \in V \setminus K. \tag{24}$$

3.1.3 Local Search Problems

A *local search problem* $\Pi = (\mathcal{I}, S, N, f)$ consists of:

- a set of *instances* $\mathcal{I}$
- for every instance $I \in \mathcal{I}$:

[5] In particular, the following decision problem is NP-complete: given an (undirected, unweighted) graph G and an integer k, decide whether G has a cut of size at least k (i.e., with k edges crossing it).

– a finite set of *feasible solutions* S_I
– a *neighbourhood function* $N_I : S_I \to 2^{S_I}$
– an *objective function* $f_I : S_I \longrightarrow \mathbb{Z}$.

The goal is, given an instance I of Π, to find a locally optimal solution; that is, a feasible solution with no better neighbours. Formally, $x \in S_I$ is a *local optimum*[6] of instance I if

$$f_I(x) \leq f_I(y) \qquad \text{for all} \;\; y \in N_I(x).$$

Definition 3.1 (Polynomial Local Search – PLS)**.** The class PLS consists of all local search problems $\Pi = (\mathcal{I}, S, N, f)$ for which there exist polynomial-time algorithms $\mathcal{A}_\Pi$, $\mathcal{B}_\Pi$ and C_Π, that

$\mathcal{A}_\Pi$: given an instance $I \in \mathcal{I}$, computes an (initial) feasible solution $x^0 \in S_I$

$\mathcal{B}_\Pi$: given an instance $I \in \mathcal{I}$ and a feasible solution $x \in S_I$ of I, computes its objective value $f_I(x)$

C_Π: given an instance $I \in \mathcal{I}$ and a feasible solution $x \in S_I$ of I, verifies whether x is a local optimum and, if it's not, returns an improving neighbour $y \in N_I(x)$ with $f_I(y) < f_I(x)$.

Such a Π is then called *polynomial*[7] *local search problem.*

It is not hard to see that Local-Max-Cut is a member of PLS. The same is true for the problem PNE-Congestion of computing a PNE of an (explicitly represented[8]) congestion game, if one takes Rosenthal's potential as the objective function (see (22)) and better-response deviations as the neighbours of a given strategy profile (i.e., exactly the vertex-neighbourhood relation of the better-response graph we defined in Section 3.1.1).

Observe that Definition 3.1 naturally suggests a *standard local search* algorithm for solving problems $\Pi \in$ PLS: use $\mathcal{A}_\Pi$ to find an initial solution, and then apply C_Π repeatedly until a local optimum is reached. For the case of congestion games, for example, this corresponds exactly to BRD (Algorithm 1). Notice, however, that standard local search may require exponentially many iterations to terminate and therefore it is not, in general, an efficient method for solving local search problem. At the same time, local search problems need not be solved only by means of local-search methods: any "abstract" algorithm that computes a local optimum is, in principle, valid. For example, this is exactly what we did in Theorem 2.9 where we used a min-cost flow algorithm to compute a PNE of a symmetric network congestion game.

In other words, using essentially the same graph-theoretic abstraction as in Section 3.1.1 for congestion games, any (polynomial) local search problem can be seen as searching for a sink of a DAG[9] (which, provably, always exists). A "canonical" way of doing this, is by starting at an arbitrary node and following any path until such a sink is reached. Such paths can be exponentially long. However, this does not exclude the existence of other, more "sophisticated" or "involved" algorithms for computing a sink of a DAG in polynomial time (e.g., via the use of linear programming).

[6]Here we define optimality as *minimizing* the objective function. This is without loss, and one can handle the case of maximization simply by setting $f_I \leftarrow -f_I$.

[7]This terminology should not be confused with "polynomial-time": local search problems in PLS need not be solvable in polynomial time. As a matter of fact, the current consensus among most complexity theorists is that *no* polynomial running-time algorithms exist for solving PLS-complete (see Section 3.2) problems. To avoid such confusion, and also simplify notation, in the remaining of these notes we will simply use the term "local search problems" to refer to problems in PLS, when this causes no ambiguity.

[8]See the discussion in Section 2.4.1

[9]Acronym for "directed acyclic graph".

3.2 PLS-completeness

Definition 3.2 (PLS-reduction)**.** Let $\Pi = (\mathcal{I}, S, N, f)$, $\Pi' = (\mathcal{I}', S', N', f')$ be two local search problems. A PLS-*reduction* from Π to Π' consists of polynomial-time computable functions h, g such that

h: maps instances $I \in \mathcal{I}$ to $h(I) \in \mathcal{I}'$

g: maps, for every instance $I \in \mathcal{I}$, local optima $x' \in S'_{h(I)}$ of $h(I)$ to local optima $g(x') \in S_I$ of I.

If such a reduction exists, we say that Π is PLS-*reducible* to Π' and we denote this by $\Pi \leq_{\mathrm{PLS}} \Pi'$.

A problem $\Pi \in$ PLS is called PLS-*complete*, if all problems in PLS are PLS-reducible to it. Intuitively, a PLS-complete local search problem is "as hard as" any other local search problem.

Theorem 3.1. Local-Max-Cut *is* PLS*-complete. Furthermore, standard local search requires exponential time: formally, there exist instances and initial solutions of* Local-Max-Cut *from which* all *sequences of locally-improving moves are exponentially long.*

The following theorem formally establishes that the problem of computing a pure Nash equilibrium of an (explicitly represented) congestion game is as hard as any local search problem:

Theorem 3.2. PNE-Congestion *is* PLS*-complete.*

Proof. We have already seen that PNE-Congestion $\in$ PLS. From Theorem 3.1, it is enough to show that Local-Max-Cut $\leq_{\mathrm{PLS}}$ PNE-Congestion.

Fix an instance $G = (V, E, w)$ of the Local-Max-Cut problem. We will construct (in polynomial time) a congestion game $\mathcal{G} = (N, R, \{S_i\}_{i \in N}, \{c_r\}_{r \in R})$ and a function $K : \boldsymbol{S} \longrightarrow 2^V$, such that, if $\boldsymbol{s}^*$ is a PNE of $\mathcal{G}$ then $K(\boldsymbol{s}^*)$ is a locally-maximum cut of G. We set the players to coincide with the nodes of our original graph, and we introduce two resources, namely a "left" and a "right" one, for each edge. We allow each node to either play all "left" resources corresponding to her incident edges, or play all "right" ones. Formally, we define:

$$N = V, \quad R = \{r_e, \bar{r}_e\}_{e \in E}, \quad S_v = \{R_v, \bar{R}_v\}$$

where

$$R_v = \{r_e \mid e \in \delta(v)\}, \qquad \bar{R}_v = \{\bar{r}_e \mid e \in \delta(v)\}.$$

Here we have used the standard notation $\delta(v) = \{\{v, u\} \in E \mid u \in V\} = E(v, V)$ for all edges incident to node v in G. Finally, we set both resources that correspond to the same edge $e \in E$ to have the same cost function

$$c_{r_e}(x) = c_{\bar{r}_e}(x) = \begin{cases} 0, & \text{if } x = 1, \\ w_e, & \text{if } x = 2, \end{cases} \qquad \forall e \in E. \tag{25}$$

Notice that the above definition is complete, since it is easy to see that, due to our construction, no resource can be played by more than two nodes (namely, the end-points of its corresponding edge).

For every outcome $\boldsymbol{s}$ of $\mathcal{G}$ we create a cut K of the original graph G by partitioning the nodes according to whether they play "left" or "right" in $\boldsymbol{s}$. Formally, we define function K by

$$K(\boldsymbol{s}) = \{v \in V \mid s_v = R_v\}.$$

Let $\boldsymbol{s}^*$ be a PNE of $\mathcal{G}$. This means that, no node currently playing "left" would like to switch to "right". Notice also that, due to definition (25), all "left" resources r_e have zero cost under profile $\boldsymbol{s}^*$ unless *both* end-points of edge e play "left", in which case it has a cost of w_e. This is equivalent to

$$c_{r_e}(n_{r_e}(\boldsymbol{s}^*)) = \begin{cases} w_e, & \text{if } u, v \in K(\boldsymbol{s}^*), \\ 0, & \text{otherwise}, \end{cases} \qquad \forall e = \{u, v\} \in E,$$

so the equilibrium condition becomes

$$\sum_{e \in E(v, K(s^*))} w_e \leq \sum_{e \in E(v, V \setminus K(s^*))} w_e.$$

Observe that the quantity on the left-hand side of the above inequality is exactly the total weight of the new edges that are going to be added to the cut if we remove node v from it, while the right-hand one is the total weight of the edges that are destroyed by that move. Therefore, this inequality is equivalent to the local maximum cut condition (23).

In a totally analogous way, by considering the deviation of the nodes currently playing "right" in $\boldsymbol{s}$, we can recover the remaining local optimality condition (24), concluding the reduction. □

4 Mixed Nash Equilibria

In this chapter we focus on *mixed* Nash equilibria (see Section 1.2), and in particular we will discuss aspects related to their existence and computation. For historical reasons, throughout this chapter we assume that our games are given in *payoff-maximization* form (see Section 1.2.1).

4.1 Bimatrix Games

As discussed before (see Section 1.1.1), two-player games can be elegantly represented in a bimatrix form (A, B), where $A, B \in \mathbb{R}^{m\times n}$: the row player has m (pure) strategies, the column player n, and A, B are the payoff matrices of the row and column players, respectively.

A *mixed* strategy profile of a bimatrix game (A, B) is then simply a pair of vectors $(\boldsymbol{x}, \boldsymbol{y}) \in \Delta^m \times \Delta^n$, where, for k integer, Δ^k denotes the k-simplex

$$\Delta^k := \Delta([k]) = \left\{ \boldsymbol{x} \in [0,1]^k \;\middle|\; \sum_{i=1}^{k} x_i = 1 \right\}.$$

The definition of a (mixed) Nash equilibrium (3) can now be written as[10]

$$\begin{aligned} \boldsymbol{x}^\top A\boldsymbol{y} &\geq \bar{\boldsymbol{x}}^\top A\boldsymbol{y} && \forall \bar{\boldsymbol{x}} \in \Delta^m \\ \boldsymbol{x}^\top B\boldsymbol{y} &\geq \boldsymbol{x}^\top B\bar{\boldsymbol{y}} && \forall \bar{\boldsymbol{y}} \in \Delta^n \end{aligned}$$

In other words, both players are best-responding to each other's mixed strategies, i.e.

$$\begin{aligned} \boldsymbol{x} &\in \mathrm{BR}_1(\boldsymbol{y}) := \operatorname*{argmax}_{\bar{\boldsymbol{x}}\in\Delta^m} \bar{\boldsymbol{x}}^\top A\boldsymbol{y} \\ \boldsymbol{y} &\in \mathrm{BR}_2(\boldsymbol{x}) := \operatorname*{argmax}_{\bar{\boldsymbol{y}}\in\Delta^n} \boldsymbol{x}^\top B\bar{\boldsymbol{y}} \end{aligned}$$

4.2 Zero-Sum Games

A zero-sum game $A \in \mathbb{R}^{m\times n}$ is a bimatrix game of the form $(A, -A)$. Therefore, it is convenient to think of them as the row player being the maximizer (trying to maximize her expected payoff, given by matrix A) and the column player being the minimizer (trying to minimize her expected cost, given by matrix A). Now the Nash equilibrium conditions (3) can be written as

$$\bar{\boldsymbol{x}}^\top A\boldsymbol{y} \leq \boldsymbol{x}^\top A\boldsymbol{y} \leq \boldsymbol{x}^\top A\bar{\boldsymbol{y}} \qquad \forall \bar{\boldsymbol{x}} \in \Delta^m, \bar{\boldsymbol{y}} \in \Delta^n \tag{26}$$

Consider the perspective of the row player. What is the best that she can do, without any prior knowledge of the column player's response? She can guarantee herself a worst-case payoff of at least

$$\max_{\boldsymbol{x}\in\Delta^m} \min_{\boldsymbol{y}\in\Delta^n} \boldsymbol{x}^\top A\boldsymbol{y} =: V_1^A. \tag{27}$$

Similarly, the column player can guarantee herself a worst-case cost of at most

$$\min_{\boldsymbol{y}\in\Delta^n} \max_{\boldsymbol{x}\in\Delta^m} \boldsymbol{x}^\top A\boldsymbol{y} =: V_2^A. \tag{28}$$

[10] Here we assume column-vector notation, i.e.,

$$\boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix} \quad \text{and} \quad \boldsymbol{x}^\top = \begin{pmatrix} x_1 & x_2 & \dots & x_m \end{pmatrix}.$$

We call such mixed strategies

$$\boldsymbol{x}^* \in \operatorname*{argmax}_{\boldsymbol{x}\in\Delta^m} \min_{\boldsymbol{y}\in\Delta^n} \boldsymbol{x}^\top A\boldsymbol{y} \qquad \text{and} \qquad \boldsymbol{y}^* \in \operatorname*{argmin}_{\boldsymbol{y}\in\Delta^n} \max_{\boldsymbol{x}\in\Delta^m} \boldsymbol{x}^\top A\boldsymbol{y} \tag{29}$$

optimal strategies for the row and column player, respectively, and the corresponding payoffs V_1^A, V_2^A in (27) and (28), *optimal values.*

Theorem 4.1 (von Neumann's Minimax Theorem [von Neumann, 1928])**.** *For any matrix* $A \in \mathbb{R}^{m\times n}$,

$$\max_{\boldsymbol{x}\in\Delta^m} \min_{\boldsymbol{y}\in\Delta^n} \boldsymbol{x}^\top A\boldsymbol{y} = \min_{\boldsymbol{y}\in\Delta^n} \max_{\boldsymbol{x}\in\Delta^m} \boldsymbol{x}^\top A\boldsymbol{y}.$$

In particular, if $\boldsymbol{x}^*, \boldsymbol{y}^*$ *are optimal strategies of the row and column player, respectively, of the zero-sum game* A*, then*

$$V_1^A = V_2^A = \boldsymbol{x}^{*\top} A\boldsymbol{y}^*$$

and such optimal strategies can be computed in polynomial time (on the size of A*).*

In light of Theorem 4.1, the common value $V^A := V_1^A = V_2^A$ is called simply the *value* of game A, and any (mixed) strategy profile $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ of optimal strategies is also called *minimax pair.*

Proof of Theorem 4.1. Fix an arbitrary matrix $A = (a_{i,j}) \in \mathbb{R}^{m\times n}$. For every $\boldsymbol{x} \in \Delta^m$, $\boldsymbol{y} \in \Delta^n$ it is

$$\boldsymbol{x}^\top A\boldsymbol{y} = \sum_{i=1}^m \sum_{j=1}^n x_i y_j a_{i,j} = \sum_{j=1}^n y_j \left(\sum_{i=1}^m x_i a_{i,j}\right) = \sum_{i=1}^m x_i \left(\sum_{j=1}^n y_j a_{i,j}\right).$$

Therefore, due to the structure and the convexity of the simplices Δ^n and Δ^m, it must be that

$$\min_{\boldsymbol{y}\in\Delta^n} \boldsymbol{x}^\top A\boldsymbol{y} = \min_{j\in[n]} \sum_{i=1}^m x_i a_{i,j} \tag{30}$$

$$\max_{\boldsymbol{x}\in\Delta^m} \boldsymbol{x}^\top A\boldsymbol{y} = \max_{i\in[m]} \sum_{j=1}^n y_j a_{i,j}. \tag{31}$$

In other words, (30) tells us that, when the column player is best-responding to a given mixed strategy $\boldsymbol{x}$ of the row player, it is enough to only consider the columns $A_{\cdot,j}$ of matrix A with the smallest expected payoff $\boldsymbol{x}^\top A_{\cdot,j}$: the support of any mixed strategy $\boldsymbol{y} \in \mathrm{BR}_2(\boldsymbol{x})$ must be a subset of these columns, and all these columns must result in the same (minimum) expected payoff for the column player. A similar interpretation for the row player holds, this time via (31).

So, the optimal strategies $\boldsymbol{x}^*$ for the row player (see (29)) are exactly the (optimal) solutions to the following linear program:

$$\begin{aligned} \max \quad & v & & \qquad (32)\\ \text{s.t.} \quad & \sum_{i=1}^m x_i a_{i,j} \geq v, & & \forall j \in [n],\\ & x_1 + x_2 + \cdots + x_m = 1 & & \\ & x_i \geq 0, & & \forall i \in [m], \end{aligned}$$

and the optimal objective value of this program is V_1^A (see (27)). And similarly, optimal strategies $\boldsymbol{y}^*$ of the column player are solutions to the linear program

$$\begin{aligned} \min \quad & u & & \qquad (33)\\ \text{s.t.} \quad & \sum_{j=1}^n y_j a_{i,j} \leq u, & & \forall i \in [m],\\ & y_1 + y_2 + \cdots + y_n = 1 & & \\ & y_j \geq 0, & & \forall j \in [n]. \end{aligned}$$

Observing the linear programs (32) and (33) we can see that they are duals of each other. Also, it is not hard to see that they are both feasible: for example, $v = \min_{i,j} a_{i,j}, x_1 = x_2 = \cdots = x_m = \frac{1}{m}$ is a feasible solution to (32). Therefore, a pair $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ of optimal primal-dual solutions to (32) and (33) exists and can be computed in polynomial time. Furthermore, the corresponding values V_1^A and V_2^A of the two programs must be equal, so

$$\boldsymbol{x}^{*\top} A\boldsymbol{y}^* \leq \max_{x\in\Delta^m} \boldsymbol{x}^\top A\boldsymbol{y}^* = \min_{y\in\Delta^n} \max_{x\in\Delta^m} \boldsymbol{x}^\top A\boldsymbol{y} = V_2^A = V_1^A = \max_{x\in\Delta^m} \min_{y\in\Delta^n} \boldsymbol{x}^\top A\boldsymbol{y} = \min_{y\in\Delta^n} \boldsymbol{x}^{*\top} A\boldsymbol{y} \leq \boldsymbol{x}^{*\top} A\boldsymbol{y}^*,$$

for any pair $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ of optimal strategies. Thus, indeed $V_1^A = V_2^A = \boldsymbol{x}^{*\top} A\boldsymbol{y}^*$. □

As a corollary of the Minimax Theorem 4.1 we can deduce the existence of NE in zero-sum games (without making use of Nash's Theorem 1.1), as well as their efficient computability:

Corollary 4.1. *Every two-player zero-sum game has a (mixed) Nash equilibrium which can be computed in polynomial time. More precisely,* $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ *is a NE of a zero-sum game* $A \in \mathbb{R}^{m\times n}$ *if and only if it is a minimax pair.*

Proof. Tutorial exercise. □

4.3 Fictitious Play

Suppose that two players are repeatedly playing a bimatrix game (A, B) and assume that, at the start of every round, they both have access to the entire history of their opponent's playing behaviour. How shall they play, in order to maximize their (expected) payoffs? What is a "natural" process to model how players "learn" from the past?

One way to do this is given by the following dynamic, known as *Fictitious Play (FP)* and proposed by George W. Brown in 1951 (see [Berger, 2007]), under which each player is best-responding to the empirical distribution that averages over her opponent's past play. Formally, at every time step $t = 1, 2, \dots$, FP maintains a pair of mixed strategies $(\boldsymbol{x}^t, \boldsymbol{y}^t)$, called *beliefs*, given by

$$\boldsymbol{x}^t = \frac{1}{t}\sum_{\tau=1}^{t} e_{i_\tau}^m \quad \text{and} \quad \boldsymbol{y}^t = \frac{1}{t}\sum_{\tau=1}^{t} e_{j_\tau}^n \tag{34}$$

where, $e_{i_t}^m \in \Delta^m$ and $e_{j_t}^n \in \Delta^n$ are basis vectors (i.e., *pure* strategies of the row and column players, respectively) such that

$$e_{i_t}^m \in \mathrm{BR}_1(\boldsymbol{y}^{t-1}) \quad \text{and} \quad e_{j_t}^n \in \mathrm{BR}_2(\boldsymbol{x}^{t-1}). \tag{35}$$

For completeness, we set $\boldsymbol{x}^0$ and $\boldsymbol{y}^0$ to be the m- and n-dimensional, respectively, zero-vectors.

It is important to note here that, in general, the best-response (pure) strategies in (35) are not uniquely defined: it might well be the case that sets $\mathrm{BR}_1(\boldsymbol{y}^{t-1})$, $\mathrm{BR}_2(\boldsymbol{x}^{t-1})$ contain more than a single element, and thus a player can have many different choices for updating her beliefs at that step. In that sense, FP is actually a general learning-dynamic "scheme", rather than a specific algorithm; we have already seen something similar for the case of better-response dynamics (see Section 2.4.2).

If needed, one can bypass this ambiguity by choosing a specific tie-breaking rule; some standard choices are breaking ties lexicographically (that is, selecting the pure strategy with the smallest index) or uniformly at random (in which case our process becomes randomized). Since all the results we present in this section hold for arbitrary choices in (35), we will omit further discussion of this issue.

It turns out that, for zero-sum games, FP beliefs indeed converge to a NE. To formalize this, we first need to define the *duality gap* of a mixed strategy profile $(\boldsymbol{x}, \boldsymbol{y})$ of a zero-sum game $A \in \mathbb{R}^{m\times n}$ as

$$\psi_A(\boldsymbol{x}, \boldsymbol{y}) \coloneqq \max_{\bar{\boldsymbol{x}}\in\Delta^m} \bar{\boldsymbol{x}}^\top A\boldsymbol{y} - \min_{\bar{\boldsymbol{y}}\in\Delta^n} \boldsymbol{x}^\top A\bar{\boldsymbol{y}}. \tag{36}$$

One can show (see tutorial exercises) that the duality gap is alway nonnegative, and becomes zero exactly at NE.

Theorem 4.2 (Robinson [1951], Shapiro [1958]). *If $(\boldsymbol{x}^t, \boldsymbol{y}^t)$ is a belief sequence under FP of a zero-sum game $A \in \mathbb{R}^{m\times n}$, then*

$$\lim_{t\to\infty} \psi_A(\boldsymbol{x}^t, \boldsymbol{y}^t) = 0.$$

Furthermore,

$$\psi_A(\boldsymbol{x}^t, \boldsymbol{y}^t) = O\left(t^{-\frac{1}{n+m-2}}\right).$$

As a matter of fact, it is conjectured by Karlin [1959, p. 183] that the rate of convergence given in Theorem 4.2 can be greatly improved:

Conjecture 4.1 (Karlin's weak conjecture). Under any "natural" choice of a tie-breaking rule for the FP beliefs $(\boldsymbol{x}^t, \boldsymbol{y}^t)$ of a zero-sum game A,

$$\psi_A(\boldsymbol{x}^t, \boldsymbol{y}^t) = O\left(\frac{1}{\sqrt{t}}\right).$$

It is important to mention here that, the original *strong* version of Karlin's conjecture, stating that *all* FP beliefs converge at a rate of $O(t^{-1/2})$, has been now disproved: Daskalakis and Pan [2014] showed a lower bound of $\Omega(t^{-1/n})$ for an identity-matrix zero-sum game $A \in \mathbb{R}^{n\times n}$.

4.4 Nash's Theorem

This section is devoted to proving Nash's Theorem 1.1. We will need the following theorems from real analysis:

Theorem 4.3 (Brouwer's Fixed-Point Theorem). *Let $K \subseteq \mathbb{R}^n$ be nonempty, compact and convex. Then every continuous function $f : K \longrightarrow K$ has a* fixed point, *i.e., there exists $x^* \in K$ such that $f(x^*) = x^*$.*

Theorem 4.4 (Berge's Maximum Theorem). *Let $X \subseteq \mathbb{R}^m$, $Y \subseteq \mathbb{R}^n$ be compact sets and $f : X \times Y \longrightarrow \mathbb{R}$ be a continuous function. If function $g : Y \longrightarrow X$ with*

$$g(y) = \operatorname*{argmax}_{x\in X} f(x, y)$$

is well-defined, then it is continuous.

Fix an arbitrary (finite) payoff-maximization game $\mathcal{G} = (N, \{S_i\}_{i\in N}, \{u_i\}_{i\in N})$, and let $m_i = |S_i|$, for all $i \in N = [n]$, and $m = m_1 + m_2 + \cdots + m_n$. First notice that the set $\Delta(S_1) \times \cdots \times \Delta(S_n)$ of mixed strategy profiles of $\mathcal{G}$ is a (nonempty) compact and convex subset of $\mathbb{R}^m$, as a cartesian product of (compact and convex) simplices. For simplicity, in the following we will denote $\Delta_i := \Delta(S_i)$, for all $i \in N$, and $\Delta := \Delta_1 \times \cdots \times \Delta_n$.

Next, define the function $\boldsymbol{f} = (f_1, f_2, \ldots, f_n) : \Delta \longrightarrow \Delta$ with

$$f_i(\boldsymbol{\sigma}) := \operatorname*{argmax}_{\bar{\sigma}_i\in\Delta_i} \left(u_i(\bar{\sigma}_i, \boldsymbol{\sigma}_{-i}) - \|\bar{\sigma}_i - \sigma_i\|^2\right) \qquad \text{for all } \; i \in N, \; \boldsymbol{\sigma} \in \Delta,$$

where $\|\cdot\|$ denotes the standard Euclidean norm (in $\mathbb{R}^{m_i}$), and recall our shorthand notation $u_i(\boldsymbol{\sigma}) = \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\sigma}}[u_i(\boldsymbol{s})]$ for mixed profiles $\boldsymbol{\sigma}$. Due to Theorem 4.3, to conclude the proof of Theorem 1.1 it is enough to show that:

1. $\boldsymbol{f}$ is well-defined,
2. $\boldsymbol{f}$ is continuous, and
3. every fixed point of $\boldsymbol{f}$ is a NE of $\mathcal{G}$.

For Point 1, we need to show that, for any player i and any mixed profile $\boldsymbol{\sigma}$, function

$$g_{i,\boldsymbol{\sigma}}(\bar{\sigma}_i) \coloneqq u_i(\bar{\sigma}_i, \boldsymbol{\sigma}_{-i}) - \|\bar{\sigma}_i - \sigma_i\|^2$$

attains a *unique* maximum over Δ_i. Observe that $u_i(\bar{\sigma}_i, \boldsymbol{\sigma}_{-i})$ is a continuous and linear function of $\bar{\sigma}_i$, and $\|\bar{\sigma}_i - \sigma_i\|^2$ a continuous and *strictly* convex[11] function of $\bar{\sigma}_i$. Therefore, $g_{i,\boldsymbol{\sigma}}$ is a continuous function over the compact set $\Delta \subseteq \mathbb{R}^{m_i}$, and thus attains its maximum in Δ_i. Furthermore, this maximum is indeed unique, since $g_{i,\boldsymbol{\sigma}}$ is strictly concave and Δ_i is a convex set.

For Point 2 now, observe that, for each $i \in N$, function f_i is well-defined and given by $f_i(\boldsymbol{\sigma}) = \operatorname{argmax}_{\bar{\sigma}_i \in \Delta_i} g_{i,\boldsymbol{\sigma}}(\bar{\sigma}_i)$, where $g_{i,\boldsymbol{\sigma}}(\bar{\sigma}_i)$ is continuous in both $\bar{\sigma}_i$ and $\boldsymbol{\sigma}$, and both $\Delta_i \subseteq \mathbb{R}^{m_i}$, $\Delta \subseteq \mathbb{R}^m$ are compact. Thus, by applying Theorem 4.4 we get that f_i is continuous; this implies the continuity of $\boldsymbol{f}$.

Finally, we will establish Point 3. Let $\boldsymbol{\sigma}^* \in \Delta$ be a fixed point of function $\boldsymbol{f}$ and fix a player i and a deviation $\sigma_i' \in \Delta_i$. For an arbitrary $\lambda \in (0, 1]$, define

$$\hat{\sigma}_i = \lambda\sigma_i' + (1 - \lambda)\sigma_i^*.$$

Since $\boldsymbol{\sigma}^* \in \Delta$ is a fixed point of $\boldsymbol{f}$, it must be that

$$\sigma_i^* = f_i(\boldsymbol{\sigma}^*) = \operatorname*{argmax}_{\bar{\sigma}_i \in \Delta_i} g_{i,\boldsymbol{\sigma}^*}(\bar{\sigma}_i).$$

This means that $g_{i,\boldsymbol{\sigma}^*}(\sigma_i^*) \geq g_{i,\boldsymbol{\sigma}^*}(\hat{\sigma}_i)$, or equivalently

$$u_i(\sigma_i^*, \boldsymbol{\sigma}_{-i}^*) - \|\sigma_i^* - \sigma_i^*\|^2 \geq u_i(\hat{\sigma}_i, \boldsymbol{\sigma}_{-i}^*) - \|\hat{\sigma}_i - \sigma_i^*\|^2$$

which, due to the linearity of $u_i(\bar{\sigma}_i, \boldsymbol{\sigma}_{-i})$ with respect to $\bar{\sigma}_i$, gives

$$\begin{aligned} u_i(\sigma_i^*, \boldsymbol{\sigma}_{-i}^*) &\geq \lambda u_i\left(\lambda\sigma_i' + (1 - \lambda)\sigma_i^*, \boldsymbol{\sigma}_{-i}^*\right) - \|\lambda\sigma_i' + (1 - \lambda)\sigma_i^* - \sigma_i^*\|^2 \\ &= \lambda u_i(\sigma_i', \boldsymbol{\sigma}_{-i}^*) + (1 - \lambda)u_i(\sigma_i^*, \boldsymbol{\sigma}_{-i}^*) - \lambda^2\|\sigma_i' - \sigma_i^*\|^2. \end{aligned}$$

Dividing both sides with $\lambda > 0$, we get

$$u_i(\sigma_i', \boldsymbol{\sigma}_{-i}^*) - u_i(\sigma_i^*, \boldsymbol{\sigma}_{-i}^*) \leq \lambda\|\sigma_i' - \sigma_i^*\|^2.$$

Taking the limit as $\lambda \longrightarrow 0$, we can finally deduce that

$$u_i(\sigma_i', \boldsymbol{\sigma}_{-i}^*) \leq u_i(\sigma_i^*, \boldsymbol{\sigma}_{-i}^*),$$

establishing that $\boldsymbol{\sigma}^*$ is indeed a NE of $\mathcal{G}$.

[11] Consider a convex subset $S \subseteq \mathbb{R}^n$. A function $f : S \longrightarrow \mathbb{R}$ is called *strictly convex* if $f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1-\lambda)f(y)$ for all $x, y \in S$ with $x \neq y$ and for all $\lambda \in (0, 1)$.

5 Correlated Equilibria

5.1 Definitions

Fix a (cost-minimization) game $\mathcal{G} = (N, \{S_i\}_{i \in N}, \{C_i\}_{i \in N})$. An (arbitrary joint) distribution $\boldsymbol{\sigma} \in \Delta(\boldsymbol{S})$ over the outcomes $\boldsymbol{S} = S_1 \times \cdots \times S_n$ will be called a *correlated strategy profile* of $\mathcal{G}$. The special case of correlated strategy profiles that are also product distributions, corresponds exactly to the notion of *mixed* strategy profiles (see Section 1.1).

Definition 5.1 (Correlated Equilibria)**.** A correlated strategy profile $\boldsymbol{\sigma} \in \Delta(\boldsymbol{S})$ of a (cost-minimization) game $\mathcal{G}$ is a *correlated equilibrium (CE)* if, for any player $i \in N$ and any *swap function* $\delta_i : S_i \longrightarrow S_i$:

$$\mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\sigma}} \left[C_i(\boldsymbol{s}) \right] \leq \mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\sigma}} \left[C_i(\delta_i(s_i), \boldsymbol{s}_{-i}) \right].$$

A standard, intuitive interpretation of Definition 5.1 is the following: there exists a trusted third party (called the *mediator*) that will randomly draw an outcome $\boldsymbol{s}$ (from the joint distribution $\boldsymbol{\sigma}$) and then will confidentially make a recommendation to each player i to follow strategy s_i; this suggestion is nonbinding, and players are free to deviate and play according to an alternative plan, given by δ_i. However, the notion of CE implies exactly that no player has an incentive to do so.

An alternative definition of CE is given in the following theorem:

Theorem 5.1. *A correlated strategy profile $\boldsymbol{\sigma}$ is a CE of a game $\mathcal{G}$ if and only if, for any player i and any strategies $s_i, s_i' \in S_i$,*

$$\mathbb{E}_{\boldsymbol{s}_{-i} \sim \boldsymbol{\sigma}_{-i} | s_i} \left[C_i(s_i, \boldsymbol{s}_{-i}) \right] \leq \mathbb{E}_{\boldsymbol{s}_{-i} \sim \boldsymbol{\sigma}_{-i} | s_i} \left[C_i(s_i', \boldsymbol{s}_{-i}) \right],$$

where $\boldsymbol{\sigma}_{-i}|s_i$ denotes[12] *the distribution of profiles $\boldsymbol{s}_{-i}$ induced by $\boldsymbol{\sigma}$ when conditioned on the strategy of player i being fixed to s_i.*

One can show (see tutorials) that the notion of CE generalizes that of NE and, furthermore, this inclusion is in general strict. In particular, due to Nash's Theorem 1.1, this implies that all games have a CE.

Example 5.1 (Traffic light game)**.** Consider the following 2×2 (cost-minimization) bimatrix game:

	Stop	*Go*
Stop	1, 1	1, 0
Go	0, 1	100, 100

(a) Costs

	Stop	*Go*
Stop	$p_{1,1}$	$p_{1,2}$
Go	$p_{2,1}$	$p_{2,2}$

(b) A correlated strategy profile

Table 6: The traffic light game

The cost matrices of the players are shown in Table 6a. A correlated strategy profile of this game is just a probability distribution over its four outcomes, and thus it can be represented via a 2×2 matrix of probabilities, as shown in Table 6b, such that $p_{1,1} + p_{1,2} + p_{2,1} + p_{2,2} = 1$.

It is not hard to see that the (pure) strategy profiles (*Stop*, *Go*) and (*Go*, *Stop*) are pure NE of the game, corresponding to the correlated profiles

$$\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \qquad \text{and} \qquad \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix},$$

[12]Formally, if $f_{\boldsymbol{\sigma}}$ is the density function of $\boldsymbol{\sigma}$, then the density of $\boldsymbol{\sigma}_{-i}|s_i$ is given by $f_{\boldsymbol{\sigma}_{-i}|s_i}(\boldsymbol{s}_{-i}) := f_{\boldsymbol{\sigma}}(s_i, \boldsymbol{s}_{-i}) / \sum_{\bar{\boldsymbol{s}}_{-i} \in \boldsymbol{S}_{-i}} f_{\boldsymbol{\sigma}}(s_i, \bar{\boldsymbol{s}}_{-i})$.

respectively. The game has also the mixed NE $\left((\frac{99}{100}, \frac{1}{100}), (\frac{99}{100}, \frac{1}{100})\right)$, which in correlated form can be written as

$$\begin{pmatrix} \frac{99^2}{10000} & \frac{99}{10000} \\ \frac{99}{10000} & \frac{1}{10000} \end{pmatrix}.$$

We will next show that

$$\begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix}$$

is a CE of the game. Notice that this can not be a NE of the game, since it is *not* (see tutorials) a product distribution. Using the notation from Table 6b, we can write the CE conditions from Theorem 5.1 as

$$\text{Player 1} \quad \begin{cases} 1 \cdot \frac{p_{1,1}}{p_{1,1}+p_{1,2}} + 1 \cdot \frac{p_{1,2}}{p_{1,1}+p_{1,2}} \leq 0 \cdot \frac{p_{1,1}}{p_{1,1}+p_{1,2}} + 100 \cdot \frac{p_{1,2}}{p_{1,1}+p_{1,2}} & (s_1 = \mathit{Stop}) \\ 0 \cdot \frac{p_{2,1}}{p_{2,1}+p_{2,2}} + 100 \cdot \frac{p_{2,2}}{p_{2,1}+p_{2,2}} \leq 1 \cdot \frac{p_{2,1}}{p_{2,1}+p_{2,2}} + 1 \cdot \frac{p_{2,2}}{p_{2,1}+p_{2,2}} & (s_1 = \mathit{Go}) \end{cases}$$

$$\text{Player 2} \quad \begin{cases} 1 \cdot \frac{p_{1,1}}{p_{1,1}+p_{2,1}} + 1 \cdot \frac{p_{2,1}}{p_{1,1}+p_{2,1}} \leq 0 \cdot \frac{p_{1,1}}{p_{1,1}+p_{2,1}} + 100 \cdot \frac{p_{2,1}}{p_{1,1}+p_{2,1}} & (s_2 = \mathit{Stop}) \\ 0 \cdot \frac{p_{1,2}}{p_{1,2}+p_{2,2}} + 100 \cdot \frac{p_{2,2}}{p_{1,2}+p_{2,2}} \leq 1 \cdot \frac{p_{1,2}}{p_{1,2}+p_{2,2}} + 1 \cdot \frac{p_{2,2}}{p_{1,2}+p_{2,2}} & (s_2 = \mathit{Go}) \end{cases}.$$

It is easy to check that $p_{1,2} = p_{2,1} = \frac{1}{2}, p_{1,1} = p_{2,2} = 0$ satisfy the system of inequalities above.

The particular form of the inequalities occurring in the characterization of the CE in Example 5.1, is not an accident. The problem of computing a CE of a game can be reduced to solving a linear programming feasibility problem, over the probabilities assigned to the different outcomes. This implies the following positive computational result:

Theorem 5.2. *A correlated equilibrium of an (explicitly represented, finite) game can be computed in polynomial time.*

Proof. See tutorials. □

The following notion is a strict (see tutorials) generalization of CE:

Definition 5.2 (Coarse Correlated Equilibria)**.** A correlated strategy profile $\boldsymbol{\sigma} \in \Delta(S)$ of a (cost-minimization) game $\mathcal{G}$ is a *coarse correlated equilibrium (CCE)* if, for any player $i \in N$ and any deviation $s_i' \in S_i$:

$$\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\sigma}}\left[C_i(\boldsymbol{s})\right] \leq \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\sigma}}\left[C_i(s_i', \boldsymbol{s}_{-i})\right].$$

Notice that the CEE conditions in Definition 5.2 corresponds to the special case of the CE conditions in Definition 5.1, if one considers only constant swap functions δ_i.

5.2 No-Regret Learning

Consider the following dynamic decision-making setting: there is a *time horizon* $T \in \mathbb{N}_{\geq 1}$ and a finite set of *actions* A. Let $|A| =: k \geq 2$. In every time step $t = 1, 2, \dots, T$:

1. the *decision maker* picks a probability distribution p^t over the set A
2. the *adversary* responds with choosing a cost function $c^t : A \longrightarrow [0, 1]$
3. an action a^t is drawn from distribution p^t; the decision maker incurs a cost of $c^t(a^t)$

An *online algorithm* $\mathcal{A}$ computes the probability distributions p^t for the decision maker at Step 1 above, for every time step $t = 1, \dots, T$, taking as input the time horizon T, and the action set A, but also having access to the previously observed costs $\{c^\tau\}_{\tau\in[t-1]}$; however, it has no knowledge of future information, and in particular of the current choice c^t of the adversary (at Step 2). The adversary, on the other hand, has a single advantage: when computing his response c^t (at Step 3), in addition to the history $\{p^\tau\}_{\tau\in[t-1]}$, he has access to the online algorithm's *current* strategy p^t as well.

The (time-averaged, external) *regret* of a sequence of actions $\boldsymbol{a} = \{a^t\}_{t\in[T]}$ with respect to a sequence of cost functions $\boldsymbol{c} = \{c^t\}_{t\in[T]}$ is defined as

$$\mathcal{R}(\boldsymbol{a}, \boldsymbol{c}) \coloneqq \frac{1}{T}\left[\sum_{t=1}^{T} c^t(a^t) - \min_{a\in A}\sum_{t=1}^{T} c^t(a)\right]. \tag{37}$$

Intuitively, this notion captures how far away (per step) the cost of our actions $\boldsymbol{a}$ is from the cost we could have incurred if we had full knowledge of the adversarial cost choices $\boldsymbol{c}$ in advance, but we had to commit to a *fixed* action $a \in A$ from the start.

An online algorithm $\mathcal{A} = \{p^t\}$ (with respect to a fixed action set A) is said to have *no regret* if its expected regret vanishes to zero. To formalize this, we define the regret of algorithm $\mathcal{A}$ (with respect to time horizon T) to be its worst-case expected regret over all adversarial cost sequences $\boldsymbol{c} = \{c^t\}_{t\in T}$, i.e.,

$$\mathcal{R}^T_{\mathcal{A}} \coloneqq \sup_{\boldsymbol{c}} \mathcal{R}_{\mathcal{A}}(\boldsymbol{c}) \quad \text{with} \quad \mathcal{R}_{\mathcal{A}}(\boldsymbol{c}) \coloneqq \mathbb{E}_{\boldsymbol{a}\sim\boldsymbol{p}}\left[\mathcal{R}(\boldsymbol{a},\boldsymbol{c})\right] = \frac{1}{T}\left[\sum_{t=1}^{T}\sum_{a\in A} p^t(a)c^t(a) - \min_{a\in A}\sum_{t=1}^{T} c^t(a)\right],$$

where $\boldsymbol{p} = p^1 \times \cdots \times p^T$ denotes the joint (product) distribution of the realized action sequence $\boldsymbol{a} = \{a^t\}_{t\in[T]}$. Then, algorithm $\mathcal{A}$ has no regret, if $\lim_{T\to\infty} \mathcal{R}^T_{\mathcal{A}} = 0$. It is not hard to see that $\mathcal{R}^T_{\mathcal{A}} \in [0, 1]$ for any algorithm $\mathcal{A}$ and any time horizon T.

It is not immediately obvious that no-regret algorithms even exist. When trying to minimize regret, one natural strategy is to choose, at every step t, the best action with respect to the cost functions experienced so far. Formally, this corresponds to the *deterministic*[13] online algorithm, known as *Follow-the-Leader (FTL)*, that sets p^t to be the distribution that assigns full probability 1 to an action in $\operatorname{argmin}_{a\in A}\sum_{\tau=1}^{t-1} c^\tau(a)$. Unfortunately, though, it turns out that the FLT algorithm has expected regret bounded away from 0. As a matter of fact, the following result establishes that randomness is necessary to achieve vanishing regret:

Theorem 5.3. *Any deterministic online algorithm for a decision-making setting with k actions, has regret at least $1 - \frac{1}{k}$.*

Proof. Fix an arbitrary deterministic online algorithm $\mathcal{A}$ over an action set $A = \{a_1, a_2, \dots, a_k\}$ and let a_{i_t}, where $i_t \in [k]$, be the action on which distribution p^t of $\mathcal{A}$ assigns its full mass. Then, for any time step t, let the adversarial cost functions be given by

$$c^t(a) = \begin{cases} 1, & \text{if } a = a_{i_t} \\ 0, & \text{otherwise,} \end{cases} \qquad \text{for all } a \in A.$$

Then, the total cost experienced by algorithm $\mathcal{A}$ is $\sum_{t=1}^{T} c^t(a_{i_t}) = T$. On the other hand, summing over all actions and all times steps, we get that also $\sum_{i=1}^{k}\sum_{t=1}^{T} c^t(a_i) = T$, and thus there has to exists an action $a_j \in A$ with total cost at most $\sum_{t=1}^{T} c^t(a_j) \le \frac{T}{k}$.

The above analysis shows that, for any time horizon T,

$$\mathcal{R}^T_{\mathcal{A}} \ge \frac{1}{T}\left[\sum_{t=1}^{T} c^t(a_{i_t}) - \sum_{t=1}^{T} c^t(a_j)\right] \ge \frac{1}{T}\left(T - \frac{T}{k}\right) = 1 - \frac{1}{k}.$$

□

[13] Determinism here refers to the property that each p^t is a single-point-mass distribution.

5.2.1 Multiplicative Weights Update

We will next show that there exists a randomized online algorithm, known as *Multiplicative Weights Update (MWU)*, that indeed achieves no-regret.

MWU is parameterized by an "external" parameter $\eta \in [0, 1/2]$, which we will fine-tune later. Throughout its execution, MWU(η) maintains a set of *weights* $\{w^t(a)\}_{a\in A}$ over the set of actions, which is initialized at $w^1(a) = 1$ for all $a \in A$. Then, at every time step $t = 1, 2, \dots, T$ and for all actions $a \in A$, it:

- assigns probabilities $p^t(a) = \frac{w^t(a)}{\sum_{\bar{a}\in A} w^t(\bar{a})}$
- updates weights by $w^{t+1}(a) = w^t(a) \cdot (1 - \eta)^{c^t(a)}$.

Theorem 5.4. *For any action set A with $|A| = k$ and any time horizon $T \geq 4 \ln k$, there is a choice of parameter $\eta \in [0, \frac{1}{2}]$ such that the expected regret of* MWU(η) *is at most $2\sqrt{\frac{\ln k}{T}}$.*

Corollary 5.1. *Fix an action set A with $|A| = k$ and an arbitrary $\varepsilon \in (0, 1)$. MWU can achieve a regret of at most ε after at most $O\left(\frac{\log k}{\varepsilon^2}\right)$ steps.*

Proof of Theorem 5.4. For our proof we will need the following technical lemma:

Lemma 5.1. *For any $0 \leq x \leq \frac{1}{2}$ and $0 \leq y \leq 1$:*

$$-x - x^2 \leq \ln(1 - x) \leq -x \tag{38}$$

and

$$(1 - x)^y \leq 1 - xy. \tag{39}$$

Fix a time horizon T and a parameter $\eta \in (0, \frac{1}{2}]$. Let $\boldsymbol{c} = \{c^t\}_{t\in[T]}$ be an adversarially selected cost sequence for MWU(η). For convenience, for any step $t = 1, 2, \dots$ we define

$$W^t = \sum_{a\in A} w^t(a) \qquad \text{and} \qquad \ell^t = \sum_{a\in A} p^t(a)c^t(a) = \frac{1}{W^t} \sum_{a\in A} w^t(a)c^t(a).$$

Then, the expected total cost of MWU(η) and the cost of the optimal fixed strategy are given by

$$L^T = \sum_{t=1}^{T} \ell^t \qquad \text{and} \qquad \mathrm{OPT}^T = \min_{a\in A} \sum_{t=1}^{T} c^t(a),$$

and thus the regret can be expressed as

$$\mathcal{R}_{\mathrm{MWU}(\eta)}(\boldsymbol{c}) = \frac{1}{T}\left(L^T - \mathrm{OPT}^T\right).$$

Our goal would be to show that

$$L^T \leq (1 + \eta)\mathrm{OPT}^T + \frac{\ln k}{\eta}. \tag{40}$$

The proof of the theorem can be then established from (40) by letting

$$T \geq 4 \ln k \qquad \text{and} \qquad \eta = \sqrt{\frac{\ln k}{T}} \leq \frac{1}{2},$$

since then we would have

$$\mathcal{R}^T_{\mathrm{MWU}(\eta)} \leq \frac{1}{T}\left[(1+\eta)\mathrm{OPT}^T + \frac{\ln k}{\eta} - \mathrm{OPT}^t\right] = \eta\frac{\mathrm{OPT}^T}{T} + \frac{\ln k}{T\eta} \leq \eta + \frac{\ln k}{T\eta} = 2\sqrt{\frac{\ln k}{T}}.$$

To prove (40), first observe that for any step $t = 1, 2, \dots, T$:

$$\begin{aligned}
W^{t+1} &= \sum_{a\in A} w^t(a)(1-\eta)^{c^t(a)} \\
&\leq \sum_{a\in A} w^t(a)(1-\eta c^t(a)), && \text{from (39),} \\
&= W^t - \eta \sum_{a\in A} w^t(a)c^t(a) \\
&= W^t - \eta W^t \ell^t \\
&= W^t(1-\eta\ell^t).
\end{aligned}$$

Taking the product for all steps $t = 1, \dots, T$ we get:

$$W^{T+1} \leq W^1 \prod_{t=1}^{T}(1-\eta\ell^t) = k\prod_{t=1}^{T}(1-\eta\ell^t). \tag{41}$$

On the other hand, we can lower-bound the same quantity by:

$$\begin{aligned}
W^{T+1} &\geq \max_{a\in A} w^{T+1}(a) \\
&= \max_{a\in A}\left[w^1(a)\prod_{t=1}^{T}(1-\eta)^{c^t(a)}\right] \\
&= \max_{a\in A}\left[(1-\eta)^{\sum_{t=1}^T c^t(a)}\right] \\
&= (1-\eta)^{\min_{a\in A}\sum_{t=1}^T c^t(a)} \\
&= (1-\eta)^{\mathrm{OPT}^T}.
\end{aligned}$$

Combining this with inequality (41) we see that

$$(1-\eta)^{\mathrm{OPT}^T} \leq k\prod_{t=1}^{T}(1-\eta\ell^t)$$

and by taking (natural) logarithms of both sides:

$$\mathrm{OPT}^T\ln(1-\eta) \leq \ln(k) + \sum_{t=1}^{T}\ln(1-\eta\ell^t).$$

Using the logarithm bounds from (38) we can simplify this to

$$-(\eta+\eta^2)\mathrm{OPT}^T \leq \ln(k) - \eta\sum_{t=1}^{T}\ell^t = \ln(k) - \eta L^T,$$

which is indeed equivalent to the desired (40). □

It turns out that the regret bound in Theorem 5.4 is (asymptotically) tight. In other words, MWU actually achieves the fastest possible regret decrease rate:

Theorem 5.5. *The regret of any online algorithm (over a set of k actions and time horizon T) is $\Omega\left(\sqrt{\frac{\log k}{T}}\right)$.*

Proof. See tutorials. □

5.3 No-Regret Game Dynamics

Fix a cost-minimization game $\mathcal{G} = (N, \{S_i\}_{i \in N}, \{C_i\}_{i \in N})$ and assume that $C_i(\boldsymbol{s}) \in [0, 1]$ for all players i and outcomes $\boldsymbol{s}$.

Consider the following decision-making process (see Section 5.2), known as *no-regret dynamics*, where each player $i \in N$ uses an arbitrary no-regret online algorithm $\mathcal{A}_i$ in order to repeatedly compute probability distributions over her strategies:

Algorithm 2 No-Regret Dynamics

Input: Game $\mathcal{G} = (N, \{S_i\}, \{C_i\})$; no-regret algorithms $\{\mathcal{A}_i\}_{i \in N}$; time horizon T
Output: A correlated strategy profile $\boldsymbol{\sigma}$ of $\mathcal{G}$

for $t = 1, \ldots, T$ **do**
  for $i \in N$ **do**
    Player i uses $\mathcal{A}_i$ to compute a distribution p_i^t over the action set S_i
  end for
  Define mixed profile $\boldsymbol{\sigma}^t = \prod_{i \in N} p_i^t$
  for $i \in N$ **do**
    The adversary presents player i with cost function

$$c_i^t(s_i) = \mathbb{E}_{\boldsymbol{s}_{-i} \sim \boldsymbol{\sigma}_{-i}^t}\left[C_i(s_i, \boldsymbol{s}_{-i})\right] \qquad \text{for all } \ s_i \in S_i$$

  end for
end for
return Correlated profile $\boldsymbol{\sigma} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\sigma}^t$

Intuitively, the joint distribution $\boldsymbol{\sigma}$ computed at the end of Algorithm 2, captures the time-averaged history of joint play under no-regret dynamics. The following theorem formalizes the fact that no-regret dynamics actually converge (fast) to CCE:

Theorem 5.6. *If at the end of the execution of Algorithm 2 the regret of every player is at most $\varepsilon > 0$, then profile $\boldsymbol{\sigma}$ is an* ε-approximate coarse correlated equilibrium (ε-CCE)*, i.e.*

$$\mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\sigma}}\left[C_i(\boldsymbol{s})\right] \le \mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\sigma}}\left[C_i(s_i', \boldsymbol{s}_{-i})\right] + \varepsilon \qquad \textit{for all } \ i \in N, \ s_i' \in S_i.$$

Corollary 5.2. *No-regret dynamics via MWU compute an ε-CCE after at most $O\left(\frac{k}{\varepsilon^2}\right)$ iterations, where* $k = \max_{i \in N} |S_i|$.

Proof of Theorem 5.6. Fix any player $i \in N$ and deviation $s_i' \in S_i$. Then:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\sigma}}\left[C_i(\boldsymbol{s})\right] &= \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{s} \sim \boldsymbol{\sigma}^t}\left[C_i(\boldsymbol{s})\right] \\
&= \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{s_i \sim p_i^t}\left[\mathbb{E}_{\boldsymbol{s}_{-i} \sim \boldsymbol{\sigma}_{-i}^t}\left[C_i(s_i, \boldsymbol{s}_{-i})\right]\right] \\
&= \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{s_i \sim p_i^t}\left[c_i^t(s_i)\right] \\
&\le \varepsilon + \frac{1}{T}\min_{s_i \in S_i}\sum_{t=1}^{T} c_i^t(s_i), && \text{since } \ \mathcal{R}_{\mathcal{A}_i}^T \le \varepsilon,
\end{aligned}$$

$$
\begin{aligned}
&\leq \varepsilon + \frac{1}{T}\sum_{t=1}^{T} c_i^t(s_i') \\
&= \varepsilon + \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{s}_{-i}\sim\boldsymbol{\sigma}_{-i}^t}\left[C_i(s_i', \boldsymbol{s}_{-i})\right] \\
&= \varepsilon + \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\sigma}}\left[C_i(s_i', \boldsymbol{s}_{-i})\right].
\end{aligned}
$$

□

For the special case of zero-sum games, it turns out that no-regret learning can provide convergence to the stricter set of NE. More precisely, the time-averaged history of each player's no-regret dynamics, converges to an optimal strategy:

Theorem 5.7. *Fix a zero-sum game $A \in [0,1]^{m\times n}$. If at the end of the execution of Algorithm 2 the regret of every player is at most $\varepsilon > 0$, then the mixed strategies $\hat{\boldsymbol{x}} = \frac{1}{T}\sum_{t=1}^{T} p_1^t$ and $\hat{\boldsymbol{y}} = \frac{1}{T}\sum_{t=1}^{T} p_2^t$ have a duality gap (see (36)) of at most*

$$\psi_A(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \leq 2\varepsilon.$$

Proof. Since the row player has regret at most ε, it must be that

$$\frac{1}{T}\left[\sum_{t=1}^{T}(p_1^t)^\top(-A)p_2^t - \min_{i\in[m]}\sum_{t=1}^{T}(-A_{i,\cdot})(p_2^t)\right] \leq \varepsilon$$

and thus

$$\max_{\boldsymbol{x}\in\Delta^m} \boldsymbol{x}^\top A\hat{\boldsymbol{y}} = \max_{i\in[m]} A_{i,\cdot}\hat{\boldsymbol{y}} = \max_{i\in[m]}\left[A_{i,\cdot}\left(\frac{1}{T}\sum_{t=1}^{t} p_2^t\right)\right] \leq \frac{1}{T}\sum_{t=1}^{T}(p_1^t)^\top A p_2^t + \varepsilon.$$

In a totally analogous way, for the column player we can get that

$$\min_{\boldsymbol{y}\in\Delta^n} \hat{\boldsymbol{x}}^\top A\boldsymbol{y} \geq \frac{1}{T}\sum_{t=1}^{T}(p_1^t)^\top A p_2^t - \varepsilon.$$

Subtracting the two inequalities we finally get the desired

$$\max_{\boldsymbol{x}\in\Delta^m} \boldsymbol{x}^\top A\hat{\boldsymbol{y}} - \min_{\boldsymbol{y}\in\Delta^n} \hat{\boldsymbol{x}}^\top A\boldsymbol{y} \leq 2\varepsilon.$$

□

Notice that Theorem 5.7 provides yet another proof for the existence of NE in two-player zero-sum games, independent of that via Nash's Theorem 1.1 or Minimax Theorem 4.1.

5.4 Swap Regret

By considering the general family of swap *functions* (see Definition 5.1) in the definition of (external) regret (37), instead of just deviations to the best *fixed* strategy, we can define a stricter benchmark, known as *swap regret*:

$$\bar{\mathcal{R}}(\boldsymbol{a}, \boldsymbol{c}) := \frac{1}{T}\left[\sum_{t=1}^{T} c^t(a^t) - \min_{\delta:A\longrightarrow A}\sum_{t=1}^{T} c^t(\delta(a^t))\right]. \tag{42}$$

Obviously, for any action and cost sequences $\boldsymbol{a}, \boldsymbol{c}$ it is $\mathcal{R}(\boldsymbol{a}, \boldsymbol{c}) \leq \bar{\mathcal{R}}(\boldsymbol{a}, \boldsymbol{c})$, since external regret (37) can be seen as a special case of swap regret (42) where only *constant* swap functions are considered.

All key notions from our no-regret learning analysis in Sections 5.2 and 5.3 can be generalized, in the natural way, to no-swap-regret learning. In particular, using no-regret algorithms as a "black-box", we can constructively show the existence of no-swap-regret algorithms as well:

Theorem 5.8 (Blum and Mansour [2007]). *Fix a set of k actions, and let $\mathcal{A}$ be an online algorithm with (external) regret $\mathcal{R}_A^T$. Then, there exists an online algorithm $\mathcal{B}$ that makes polynomially many oracle calls to $\mathcal{A}$ and has* swap *regret of at most $\bar{\mathcal{R}}_{\mathcal{B}}^T \leq k\mathcal{R}_A^T$, for any time horizon T.*

Furthermore, given the definition of swap regret via the use of swap functions (42), in a totally analogous way to the proof of Theorem 5.6 we can now establish a natural connection between no-swap-regret learning and CE:

Remark 5.1. *The time-averaged history of joint play under no-swap-regret dynamics converges (fast) to the set of CE.*

References

U. Berger. Brown's original fictitious play. *Journal of Economic Theory*, 135(1):572–578, jul 2007. doi:10.1016/j.jet.2005.12.010.

A. Blum and Y. Mansour. From external to internal regret. *Journal of Machine Learning Research*, 8:1307–1324, 2007. ISSN 1532-4435. URL `https://www.jmlr.org/papers/volume8/blum07a/blum07a.pdf`.

S. Chien and A. Sinclair. Convergence to approximate nash equilibria in congestion games. *Games and Economic Behavior*, 71(2):315–327, Mar. 2011. doi:10.1016/j.geb.2009.05.004.

G. Christodoulou and E. Koutsoupias. The price of anarchy of finite congestion games. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 67–73, 2005. doi:10.1145/1060590.1060600.

W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. John Wiley & Sons, 1998. doi:https://doi.org/10.1002/9781118033142.

C. Daskalakis and Q. Pan. A counter-example to karlin's strong conjecture for fictitious play. In *Proceedings of the 55th Annual Symposium on Foundations of Computer Science (FOCS)*, Oct. 2014. doi:10.1109/focs.2014.10.

A. Fabrikant, C. Papadimitriou, and K. Talwar. The complexity of pure nash equilibria. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pages 604–612, 2004. doi:10.1145/1007352.1007445.

M. Garey, D. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, Feb. 1976. doi:10.1016/0304-3975(76)90059-1.

S. Ieong, R. McGrew, E. Nudelman, Y. Shoham, and Q. Sun. Fast and compact: A simple class of congestion games. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*, AAAI'05, pages 489—-494, 2005. URL `https://www.aaai.org/Papers/AAAI/2005/AAAI05-077.pdf`.

S. Karlin. *Matrix Games, Programming, and Mathematical Economics*, volume 1 of *Mathematical Methods and Theory in Games, Programming, and Economics*. Addison-Wesley, 1959. doi:10.1016/C2013-0-07880-2.

D. Monderer and L. S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124–143, 1996. doi:10.1006/game.1996.0044.

J. F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1): 48–49, jan 1950. doi:10.1073/pnas.36.1.48.

C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

J. Robinson. An iterative method of solving a game. *The Annals of Mathematics*, 54(2):296, Sept. 1951. doi:10.2307/1969530.

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

H. N. Shapiro. Note on a computation method in the theory of games. *Communications on Pure and Applied Mathematics*, 11:587–593, 1958.

J. von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100(1):295–320, Dec. 1928. doi:10.1007/bf01448847.

L. A. Wolsey. *Integer Programming*. John Wiley & Sons, 2nd edition, 2020. doi:10.1002/9781119606475.