A Course in Networks and Markets

Rafael Pass
Cornell Tech

Last updated: January 3, 2018

 ISBN:

10 9 8 7 6 5 4 3 2 1

First pre-edition: February 2017

To Shira-Perla and Isaak, det finaste jag vet.

Introduction

In this course, using tools from game theory and graph theory, we explore how network structures and network effects play a role in economic and information markets. Let us start by providing an overview through a few examples.

Markets with Network Effects: iPhone vs. Android Consider a market with two competing mobile phone brands, where the buyers are connected through a social network (see Figure 0.1). Each phone has some *intrinsic* value to a buyer, but the *actual* value of phone is affected by how many of the buyer's friends (i.e., the nodes connected to them in the social network) have the same phone—this is referred to as a *network effect*: For instance, even if a buyer prefer iPhones *in isolation* (i.e., they have a higher intrinsic value to them), they may prefer to get an Android phone (or even switch to one, if they currently have an iPhone), if enough of their friends have an Android.

Some questions that naturally arise are:

- Will there eventually be a stable solution, where everyone is happy with their phone, or will people keep switching phones?
- If we arrive at a stable solution, what will it look like? (For instance, will everyone eventually have the same phone, or can we get a market for both?)

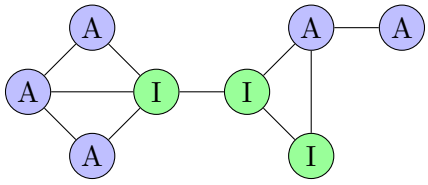


Figure 0.1: An example of a small social network for the Android/iPhone game. Nodes correspond to buyers. We draw an edge between nodes that are friends.

- If we want to market iPhones, to which "influential" individuals should we offer discounts in order to most efficiently take over the market?
- How should we set the price of a phone to best market it? (Perhaps start low and increase the price as more people buy it?)

We will study models that allow us to answer these questions. Note that this type of modeling is not only useful to study the spread of products, but can also be used to reason about the spread of (e.g., political) news or other information (or disinformation) in a social networks. For instance, I may post a news article on my Facebook wall if many of my friends do it.

The Role of Beliefs. In the example above, to market a phone, it may suffice that enough people *believe* that their friends will buy the phone. If people believe that their friends will buy the phone (accomplished e.g., by advertising), their *perceived value* of the phone will increase, and they will be more likely to buy it—we get a "*self-fulfilling prophecy*". As we shall see, in some situations, it may even be enough that there exist people who *believe there exist people who believe* (etc.) that enough people will buy the phone for this effect to happen—that is, so-called *higher-level beliefs* can have a large impact. We will study models for discussing and analyzing such higher-level beliefs—perhaps surprisingly, networks will prove useful also for modeling higher-level beliefs. We shall next use these models to shed light on the emergence of *bubbles* and *crashes* in economic markets.

More generally, we will discuss how crowds process information and how and why the following phenomena can occur:

- *The wisdom of crowds:* In some situations, the aggregate behavior of a group can give a significantly better estimate of the "truth" than any one individual (e.g., prediction teams outperforming single analysts in elections).

- *The foolishness of crowds:* In other situations, "misinformation" can be circulated through a social network in "information cascades" (e.g., the spread of urban legends/"fake news" through a social network).

Matching Markets, Auctions and Voting. Let us finally consider a quite different type of market. Assume we have three people A, B, C and three houses called H_1, H_2, H_3. The people may have some constraints on what houses are acceptable to them; we can depict the situation using a graph as shown in Figure 0.2. Can we find a "matching" (i.e., pairing) between people and houses that respects these acceptability constraints? In this simple

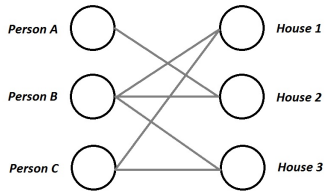


Figure 0.2: The "acceptability" graph in a matching problem. We draw a edge between a person and a house if the person finds the house acceptable.

example, it is easy to see that A can be matched with H_2, B with H_1, and C with H_3; we will study algorithms for solving this problem, and more generally, understanding when a matching where everyone gets matched exists.

Consider, now, a variant of this problem where everyone finds *all* houses acceptable, but everyone prefers H_1 to H_2, and H_2 to H_3. How should we now assign houses to people? Note that no matter how we assign houses to people, 2 people will be unhappy with their house (in the sense that they would have preferred a different house)!

The key for overcoming this problem is to assign prices to the three houses. This gives rise to the following questions:

- Can we set prices for these three houses so that everyone can be matched with their *most preferred* house (taking into account the price of the house)? Indeed, we will show that such, so-called, "market-clearing prices" are guaranteed to exist (and the hope is that the market will converge on these prices over time).

- Can we design a mechanism that incentivizes people to *truthfully* report how much each house is worth to them, so that we can assign houses to people in a way that maximizes the total "happiness" of all the people? Indeed, we shall study the Vickrey-Clark-Groves (VCG) *auction* mechanism that enables doing this.

We next note that the methods we use to provide answers to the above questions form the basis for the auction mechanisms used in *sponsored search*, where advertisers bid on "slots" for sponsored results in Internet search queries (and need to pay to get their advertisement displayed)—in this context, the goal is to find a matching between advertisers and slots.

We will also consider the "standard" (non-sponsored) web search problem: think of it as matching webpages with "slots" in the search ranking, but

the difference with the sponsored search problem is that now there are *no payments*. We will discuss the "relevance" algorithms used by search engines (e.g., Google's PageRank algorithm) to determine how (non-paying) pages returned by a search should be ordered. Here, the network structure of the Internet will be the central factor for computing a relevance score. The basic idea behind these methods is to implement a *voting* mechanism whereby other pages "vote" for each page's relevance by linking to it.

Finally, we will discuss voting schemes (e.g., for presidential elections) more generally, and investigate why such schemes typically are susceptible to "strategic voting", where voters are incentivized to not truthfully report their actual preferences (for instance, if your favorite candidate in the US presidential election is a third-party candidate, you may be inclined to vote for your second choice).

Outline of the course. The course is divided into four main parts.

- *Part 1: Games and Graphs.* In Part 1, we first introduce basic concepts from game theory (the study of how *rational agents*, trying to maximize their utility, interact) and graph theory (the study of *graphs*, mathematical constructs used to model networks of interconnected nodes). We then use concepts from both to analyze "networked coordination games" on social networks—such games provide a framework for analyzing situations similar to the Android/iPhone game discussed above.

- *Part 2: Markets on Networks.* In Part 2, we begin by introducing some more advanced algorithms for exploring graphs, and then use these algorithms to explore various different types of markets on networks (including e.g., the above-discussed market for matching houses to people).

- *Part 3: Mechanisms for Networks.* In Part 3, we discuss mechanisms for taming the above-mentioned auctions, web search, voting, and matching, problems.

- *Part 4: The Role of Beliefs.* Finally, in Part 4, we discuss various ways of modeling people's beliefs and knowledge, and explore how people's beliefs (and the above-mentioned higher-level beliefs) play a role in auctions and markets.

Comparison with Easley-Kleinberg. The topics covered here, as well as the whole premise of using a combination of game-theory and graph-theory

to study markets, is heavily inspired by Easley and Kleinberg's (EK) beautiful book "Networks, Crowds and Markets" [EK10]. However, whereas our selection of topics closely follows EK, our treatment of many (but not all) of the topics is somewhat different. In particular, our goal is to provide a formal treatment, with full proofs, of the simplest models exhibiting the above-described phenomena, *while only assuming that people are "rational agents", acting in a way that maximizes some internal"utility" function.* As such, we are also covering fewer topics than EK: in particular, we are simply assuming that the network structure (e.g., the social-network in the first example) is *exoneously* given—we do not consider how this network is formed, and do not discuss properties of it. There is a number of beautiful models and results regarding the structure of social networks (e.g., the Barabasi-Albert preferential attachment model [BA99], Watts-Strogatz small worlds model [WS98], and Kleinberg's decentralized search model [Kle00]), which are discussed in depth in EK. We also do not discuss specific diffusion models (e.g., SIR/SIS epidemic models) for modeling the spread of diseases in a social network; instead, we focus only on studying diffusion in a game-theoretic setting where agents rationally decide whether to, for instance, adopt some technology (as in the first example above).

Finally, we only rarely discuss behavioral or sociological experiments or observations (whereas EK discusses many intriguing such experiments and observations)—in a sense, we mostly focus on the mathematical and computational models. As such, we believe that a reader of these notes should read EK for the behavioral/sociological context.

Prerequisites. We will assume basic familiarity with probability theory; a primer on probability theory, which covers all the concepts and results needed to understand the material in the course, is provided Appendix A. Basic notions in computing, such as running-time of algorithms, will also be useful (but the material should be understandable also without it). Finally, we assume a basic level of mathematical maturity (e.g., comfort with definitions and proofs).

Intended audience. Most of the material in these notes is appropriate for a Master's level, or advanced undergraduate-level, course in Networks and Markets. We have also included some more advanced material (marked as such) which could be included in a introductory Ph.D. level course.

Acknowledgements. I am extremely grateful to Andrew Morgan who was the teaching assistant for CS 5854 in 2016 and 2017; Andrew edited and typeset my first version of these notes, created all the figures in the notes, came up with many of the examples in the figures, and found many mistakes and typos. Andrew also came up with many amazing HW problems! Thank you so very much!

I am also very grateful to the students of CS 5854 in 2016 and 2017, as well as Antonio Marcedone and Thodoris Lykouris who provided useful feedback on the notes. Finally, I am extremely grateful to Jon Kleinberg, Joseph Halpern and Éva Tardos for many helpful discussions.

Contents

Part I

Games and Graphs

Chapter 1

Game Theory

In this chapter, we will develop some basic tools for reasoning about strategic agents attempting to maximize their utility. We begin with one of the most basic and well-known game in the field—the **prisoner's dilemma**.

1.1 The Prisoner's Dilemma Game

Two robbers robbed a bank but were caught afterwards. They are put into separate rooms and given the choice to either *defect (D)*—accuse their partner, or *cooperate (C)*—stay silent.

- If both robbers *defect* (i.e., accuse their partner), they are both charged as guilty and given 4 years in prison.
- If both *cooperate* (i.e., remain silent), there is less evidence to charge and they will both just get 1 year in prison.
- However, if one robber *cooperates* (remains silent) and the other *defects* (accuse their partner), the cooperating one receives 10 years in prison, while the defecting one goes free.

How should the robbers act?

To study the situation, we can represent it as a **game**: we have two players (the robbers)—let us call them player 1 and 2—and each player needs to choose one of the two actions, C, D. To analyze how these players should act we also need to translate the "desirability" of each possible outcome of the game into a "score", referred to as the **utility** of the outcome (where more desirable outcomes are assigned a higher score). This is done by defining

utility functions u_1, u_2 for each of the players, where $u_i(a_1, a_2)$ denotes the utility (i.e., "score") of player i in the outcome where player 1 chose the action a_1 and player 2 chose a_2. For instance, let:

- $u_1(D, D) = u_2(D, D) = -4$ (since both players get 4 years of jail)
- $u_1(C, D) = -10$, $u_2(C, D) = 0$ (since player 1 gets 10 years of prison and player 2 goes free)
- $u_1(D, C) = 0$, $u_2(D, C) = -10$ (since player 2 gets 10 years of prison and player 1 goes free)
- $u_1(C, C) = u_2(C, C) = -1$ (since they get 1 year in prison)

We can represent this game as a table, as follows:

1 \ 2	$(*, C)$	$(*, D)$
$(C, *)$	(-1, -1)	(-10, 0)
$(D, *)$	(0, -10)	(-4, -4)

Notice that each player gets strictly more utility by defecting rather than cooperating, *no matter what the other player does*: $u_1(D, *) > u_1(C, *)$ and $u_2(*, D) > u_2(*, C)$ regardless of what action $*$ is. Thus, one would expect both robbers/players to defect in this game, thus both ending up in prison (despite the fact that they could both have been free, if they had just both cooperated)!

We turn to formalizing the notion of a game, and subsequently this way of reasoning.

1.2 Normal-form games

We will focus on a restriced class of games, satisfying the following properties:

- Players act only once;
- Players act simultaneously (i.e. without knowledge of what the other player will do);
- The number of players and the set of actions is finite.

Despite its simplicity, this class of games will suffice for most of the applications we will be considering here. The formal definition follows.

Definition 1.1. A (finite) **normal-form game** is a tuple $\mathcal{G} = (n, A, u)$, where

- $n \in \mathbb{N}$ (the number of players);
- $A = A_1 \times A_2 \times \ldots \times A_n$ is a product of finite sets; we refer to A_i as the *action space* of player i, and refer to the product space, A, as the *outcome space*;
- $u = (u_1, \ldots, u_n)$ is a tuple of functions such that $u_i : A \to \mathbb{R}$ (that is, u_i is the utility function of player i, mapping outcomes to real numbers).

We refer to a tuple $\vec{a} = (a_1, \ldots, a_n) \in A$ of actions as an *action profile*, or *outcome*. As players act only once, the **strategy** of a player is simply the action they take, and as such we use the terms strategy and action interchangeably; the literature sometime also calls such strategies *pure strategies*. As we briefly mention below, one may also consider a more general notion of a strategy—called a *mixed strategy*—which allows a player to randomize (or "mix") over actions. Furthermore, we may also consider more general classes of games—called *extensive-form games*—where players receive some inputs or need to act several times; then the strategy of a player would be a function from the "view" of the player in the game, to actions.

Some more notational details:

- Let $[n] = [1, 2, \ldots, n]$;
- Given an action set A, let A_{-i} denote the set of actions for everyone but player i (formally, $A_{-i} = \times_{j \neq i} A_j$);
- Similarly, given an action profile $\vec{a}$, let a_{-i} denote the action profile of everyone *but* player i;
- We will use (a_i, a_{-i}) to describe the full action profile where player i plays a_i and everyone else a_{-i};
- To simplify notation, we sometimes directly specify u as a function from $A \to \mathbb{R}^n$ (i.e., a function from outcomes to a *vector* of utilities where component i specifies the utility of player i).

Formalizing the Prisoner's Dilemma. We can model the Prisoner's Dilemma game that we discussed above as a normal-form game (n, A, u), where:

- $n = 2$;
- $A_1 = A_2 = \{C, D\}$;
- u is defined as follows:
  - $u(C, C) = (-1, -1)$ (i.e., $u_1(C, C) = u_2(C, C) = -1$);
  - $u(C, D) = (-10, 0)$;
  - $u(D, C) = (0, -10)$;
  - $u(D, D) = (-4, -4)$.

1.3 Dominant Strategies

In the Prisoner's Dilemma, we argued that playing D was always the best thing to do. The notion of a *dominant strategy* formalizes this reasoning.

Definition 1.2. A **dominant strategy** for a player i in a game (n, A, u) is an action a_i such that, for all $a_i' \in A_i \setminus a_i$ and all action profiles $a_{-i} \in A_{-i}$, we have

$$u_i(a_i, a_{-i}) \geq u_i(a_i', a_{-i})$$

If this inequality is *strict* (i.e. $u_i(a_i, a_{-i}) > u_i(a_i', a_{-i})$), we call a_i a **strictly dominant strategy**.

The following claim follows from our argument above.

Claim 1.3. *Defecting (D) is a strictly dominant strategy for both players in the Prisoner's Dilemma.*

The fact that D is strictly dominant is good news in the robber situation we described: the police can put the criminals in prison (assuming they act rationally). But consider a similar game where, for instance, two countries are deciding whether to cooperate or fight one another (defect) described by the same utility function. With defect as the dominant strategy, the overall utility of $(-4, -4)$ would actually be a much worse outcome than if they decided to cooperate and get $(0.5, 0.5)$.

1.4 Iterated Strict Dominance (ISD)

In general, when a game has a strictly dominant strategy, this strategy gives a good indication of how people will actually play the game. But, not every game has a strictly dominant, or even just a dominant, strategy. To illustrate this, consider the following game.

The Up-Down Game Player 1 can choose to go up (U) or down (D); player 2 can choose to go left (L), middle (M), or right (R). The following table lists the utilities for each possible outcome:

1 \ 2	$(*, L)$	$(*, M)$	$(*, R)$
$(U, *)$	(5, 5)	(0, -10)	(5, 0)
$(D, *)$	(0, 0)	(5, -10)	(0, 5)

The first thing we notice here is that M is always a "terrible" strategy; player 2 would never have any reason to play it! In fact, we say that M is *strictly dominated* by both L and R, by the following definition:

Definition 1.4. Given a game (n, A, u), we say that $a_i \in A_i$ is **strictly dominated by** a_i' (or a_i' **strictly dominates** a_i) for a player i, if for any action profile $a_{-i} \in A_{-i}$, we have

$$u_i(a_i, a_{-i}) < u_i(a_i', a_{-i})$$

We say that $a_i \in A_i$ is (simply) **strictly dominated** for a player i if there exists some strategy $a_i' \in A_i$ that strictly dominates a_i.

It is worthwhile to note that strict dominance is *transitive*—if for player i, strategy a_i dominates b_i and b_i dominates c_i, then a_i dominates c_i. A useful consequence of this is that not all strategies can be strictly dominated (prove it!).

Now, let us return to the Up-Down game. Since M is terrible "no matter what", we should remove it from consideration. Let us thus consider the game resulting from removing this action.

1 \ 2	$(*, L)$	$(*, R)$
$(U, *)$	(5, 5)	(5, 0)
$(D, *)$	(0, 0)	(0, 5)

Now look at player 1's choices; at this point, D is strictly dominated by U and can thus be removed from consideration, resulting in the following game:

1 \ 2	$(*, L)$	$(*, R)$
$(U, *)$	(5, 5)	(5, 0)

And, finally, player 2 can rule out R, leaving us with (U, L) as the only possible rational outcome. This process is known as *iterative removal of strictly dominated strategies*, or *iterated strict dominance (ISD)*. More formally,

Definition 1.5. Given a game (n, A, u), we define the set of strategies surviving **iterated strict dominance (ISD)** as follows:

- For each player i, let $A_i^0 = A_i$.

- Next, we proceed in rounds: For each round j, for each player i, let A_i^j denote the set of strategies that are not strictly dominated if we restrict the action space to $A^{j-1} = A_1^{j-1} \times \ldots \times A_n^{j-1}$. (That is, A_i^j is obtained by taking A_i^{j-1} and removing all actions a_i for which there exists some $a_i' \in A_i^{j-1}$ such that for all $a_{-i} \in A_{-i}^{j-1}$, $u_i(a_i, a_{-i}) < u_i(a_i', a_{-i})$.)
- Continue this procedure until no more strictly dominated strategies can be removed.

Note that since not all strategies can be strictly dominated in a game, this deletion procedure always ends with a non-empty set.

ISD and Common Knowledge of Rationality Intuitively, no "rational" players should ever play a strictly dominated strategy (such as M above)—since it is stricly worse, that would be a silly thing to do. So, if everyone is rational, nobody will play strictly dominated strategies. Furthermore, if everyone *knows* that everyone is rational, then, we can effectively restrict our attention to the game where all strictly dominated strategies have been removed (as everybody knows nobody will play them), and then remove dominated strategies from this smaller game. Thus, inductively, if we have *common knowledge of rationality*—that is, everybody is rational, everybody *knows* that everybody is rational, everybody *knows* that everybody *knows* that everybody is rational, and so forth—people can only play strategies that survive ISD. Later on in the course (in Chapter 17), we will see how this statement can be formalized: in fact, we show that common knowledge of rationality *exactly* characterizes the set of strategies that survive ISD, in the sense that a strategy survives ISD if and only if it is compatible with common knowledge of rationality.

1.5 Nash Equilibria and Best-Response Dynamics

Sometimes even iterative strict dominance does not give us enough power to predict the outcome of a game. Consider the following game.

Bach-Stravinsky: a Coordination Game Two players (husband and wife) are deciding whether to go to a Bach concert (B) or a Stravinsky concert (S). The first player prefers Bach and the second Stravinsky, but they will both be unhappy if they do not go together. Formally, $u(B, B) = (2, 1)$, $u(S, S) = (1, 2)$, and $u(B, S) = u(S, B) = (0, 0)$. This is a special case of

a so-called *coordination game* where, more generally, the players get "high" utility when they "coordinate" (i.e., choose the same action), and 0 otherwise.

Note that there are no dominant or dominated strategies in this game. So how can we predict what will happen? The classic way to deal with such a situation is to find *equilibrium* states, or action profiles with the property no player can increase their utility by changing their action. For instance, (B, B) is an equilibrium in this game: if player 1 switched they would lose 2 in utility, whereas player 2 would also lose 1 in utility by switching. Symmetrically, (S, S) is an equilibrium state as well.

Pure-Strategy Nash Equilibrium (PNE) We now turn to formalizing this notion through what is called a Nash equilibrium.

Definition 1.6. Given a game (n, A, u), a **Pure-strategy Nash equilibrium (PNE)** is a profile of action $\vec{a} \in A$ such that for each player i and all actions $a_i' \in A_i$,

$$u_i(a_i, a_{-i}) \geq u_i(a_i', a_{-i})$$

In other words, there does not exist some player i that has a "profitable deviation": no player i can increase their own utility by *unilaterally deviating*, assuming that everyone else sticks to the equilibrium strategy.

Relating PNE and ISD Observe that (D, D) in the Prisoner's Dilemma and (U, L) in the Up-Down game are both PNEs for their respective games. The following claims shows that this was not a coincidence: PNE is a strict refinement of ISD (and thus also strict dominance, since strictly dominant strategies can never be dominated), and when ISD produces a single strategy profile, it must be a PNE.

Claim 1.7. *Every PNE survives ISD.*

Proof. Consider a PNE $\vec{a}$, and assume for contradiction that $\vec{a}$ does not survive ISD. Consider the *first* round j when some player i's action a_i gets eliminated. Then $a_{-i} \in A_{-i}^{j-1}$ (since j was the first round when some players action in $\vec{a}$ was eliminated) and, in step j, there must have been some action a_i' that strictly dominates a_i with respect to A_{-i}^{j-1}, and hence also w.r.t. a_{-i}; that is, $u_i(a_i, a_{-i}) < u_i(a_i', a_{-i})$, which contradicts the assumption that $\vec{a}$ is a PNE. ■

Claim 1.8. *If a single strategy profile survives ISD, then it is the unique PNE of the game.*

Proof. Consider some game where a unique strategy profile $\vec{a}$ survives ISD. First, note that by Claim 1.7, there can be at most one PNE in the game (since every PNE survives ISD). Let us next show that $\vec{a}$ must be the (unique) PNE. Assume for contradiction that $\vec{a}$ is *not* a PNE. That is, there exist a player i and action a_i' such that a_i' strictly dominates a_i with respect to a_{-i} (i.e. $u_i(a_i, a_{-i}) < u_i(a_i', a_{-i})$). Since a_i' did not survive the deletion process (as only $\vec{a}$ survived ISD by assumption), a_i' must have been previously deleted due to being dominated by some strategy a_i^1 which in turn was deleted by some strategy a_i^2 and so on and so forth, until we reach some strategy a_i^m that is deleted by a_i (since only a_i survives ISD). This contradicts transitivity of strict dominance (since the strategy space shrinks at each iteration, preventing a strategy strictly dominated earlier in the process from not being strictly dominated later). ■

Best responses Another way to think of PNEs is in terms of the notion of a *best response*. Given an action profile $\vec{a}$, let $BR_i(\vec{a})$— i's **best-response set**—be the set of strategies $a_i' \in A_i$ that *maximize* player i's utility given a_{-i}; that is, the set of strategies a_i' that maximize $u_i(\cdot, a_{-i})$:

$$BR_i(\vec{a}) = \arg\max_{a_i' \in A_i} u_i(a_i', a_{-i})^1$$

Think of $BR_i(\vec{a})$ as the set of strategies that player i would "want to play" if we start off in the outcome $\vec{a}$ and players $-i$ stick to their strategy in $\vec{a}$.

Let $BR(\vec{a}) = \times_{i \in [n]} BR_i(\vec{a})$ (i.e., $\vec{a} \in BR(\vec{a})$ if and only if for all $i \in [n]$, $a_i \in BR_i(\vec{a})$.) That is, we can think of $BR(\vec{a})$ as the set of strategies the players would be happy to play if they believe *everyone else* sticks to their strategy in $\vec{a}$.

The following claim is almost immediate from the definition.

Claim 1.9. *$\vec{a}$ is a PNE if and only if $\vec{a} \in BR(\vec{a})$.*

Proof. We prove each direction separately:

The "if" direction Assume for contradiction that $\vec{a} \in BR(\vec{a})$, yet there exists some player i that has a "profitable deviation" a_i' such that

$$u_i(a_i', a_{-i}) > u_i(a_i, a_{-i})$$

Then, a_i clearly does not maximize i's utility given a_{-i}; that is, $a_i \notin BR_i(\vec{a})$

[1] Recall that $\arg\max_{x \in X} f(x) = \{y \mid \forall x \in X, f(x) \leq f(y)\}$; that is, the set of values that maximize the expression $f(\cdot)$.

The "only-if" direction Assume for contradiction that $\vec{a}$ is a PNE, yet there exists some i such that $a_i \notin BR_i(\vec{a})$. Since a_i does not maximize i's utility given a_{-i}, there must exists some other strategy a_i' that "does better" than a_i (given a_{-i}); that is,

$$u_i(a_i', a_{-i}) > u_i(a_i, a_{-i})$$

So, $\vec{a}$ cannot be a PNE as player i has a "profitable deviation" a_i.

■

Despite the simplicity of this characterization, thinking in terms of best responses leads to an important insight; a PNE can be thought of a *fixed point* of the best response operator BR—a fixed-point to a "set-valued-function"[2] f is some input x such that $x \in f(x)$. As will shall see, the notion of a fixed-point will be instrumental to us throughout the course.

Best-Response Dynamics (BRD) As argued, PNE can be thought of as the stable outcomes of play—nobody wants to unilaterally disrupt the equilibrium. But how do we arrive at these equilibria? Note that even though we are considering a *single-move* game, the equilibrium can be thought of as a stable outcome of play by players that "know" each other and how the other players will play in the game (we shall formalize this statement in Chapter 17). The question is, however, how do people arrive at a state where they "know" what the other player does—how do they "learn" how to play?

A particularly natural approach for trying to find the PNE is to start at some arbitrary action profile $\vec{a}$, and then to let any player deviate by playing a best-response to what everyone else is currently doing. That is, players "myopically" believe that everyone will continue to act exactly as they did in the previous round, and best respond to this belief. We refer to this process as **best-response dynamics (BRD)** and formalize it as follows:

1. Start off with any action profile $\vec{a}$.
2. For each player i, calculate the best-response set $BR_i(\vec{a})$.
3. If $\vec{a} \in BR(\vec{a})$, then we have arrived at an equilibrium (by Claim 1.9); return $\vec{a}$.
4. Otherwise, pick *any* player i for which $a_i \notin BR_i(\vec{a})$. Replace a_i by *any* action in $BR_i(\vec{a})$. Return to step 2. (Other players' best responses could change as a result of changing a_i, so we must recalculate.)

[2] That is, a function that outputs a set of elements.

Running best-response dynamics on the Bach-Stravinsky game starting at (B, S), for instance, will lead us to the equilibrium (B, B) if player 2 switches, or (S, S) if player 1 switches. We say that BRD **converges** in a game Γ if the BRD process ends *no matter* what the starting point is, and *no matter* what player and action we pick in each step (in case there are multiple choices).

While BRD converges for many games of interest, this is not always the case. In fact, there are some games that do not even have a PNE: Consider, for instance, the Rock-Paper-Scissor's game, where the action space is $\{R, P, S\}$, a winning player gets 1, a losing player gets 0, and both get 0 in the case of a draw; clearly, there is no PNE in this game—the looser (or either player in the case of a draw) prefers to switch to an action that beats the other player. But there are even games with a *unique PNE* for which BRD fails to converge (*Exercise*: show this. *Hint*: try combining Rock-Paper-Scissors with a game that has a PNE.).

Luckily, for most of the games we will be considering, finding a PNE with BRD will suffice. Furthermore, for games for which BRD does converge, we can be more confident about the outcome actually happening in "real life"—people will eventually arrive at the PNE if they iteratively "best-respond" to their current view of the world. (Looking forward, once we have had some graph-theory background, we can provide an elegant characterization of the class of games for which BRD converge.) As a sanity check, note that in any game where each player has a strictly dominant strategy, BRD converges to the strictly dominant action profile.

Claim 1.10. *Consider an n-player game $\mathcal{G}$ with a strategy profile $\vec{a}$ such that for every player i, a_i is strictly dominant for i. Then BRD converge to $\vec{a}$ in at most n rounds.*

Proof. In each round, some player switches to its strictly dominant strategy and will never ever switch again. Thus, after at most n rounds everyone has switched to their strictly dominant action a_i. ■

1.6 A Cautionary Game: The Traveler's Dilemma

Let's end this chapter by considering a "cautionary" game, where our current analysis methods fail to properly account for the whole story—even for a game where a PNE exists and BRD converge to it. This game also serves as a nice example of BRD.

Consider the following game: Two travelers fly to China and buy identical vases while there. On the flight back, both vases are broken; the airline

company wants to reimburse them, but does not know how much the vases cost. So they put each of the travelers in separate rooms and ask them to report how much their vase cost (from $2 to $100).

- If both travelers report the same price, they both get that amount.
- If the reports of the travelers disagree, then whoever declared the lowest price v gets $v + 2$ (as a bonus for "honesty"), while the other gets $v - 2$ (as a penalty for "lying").

At first glance, it might appear that both players would simply want to declare 100. But this is not a PNE—if you declare 100, I should declare 99 and get 101 (while you only get 97)! More precisely, if player 1 best responds, we end up at the outcome $(99, 100)$. Next, player 2 will want to deviate to 98 (which gives him 100), leading to the outcome $(99, 98)$. If we continue the BRD process, we get the sequence of outcomes $(97, 98), (97, 96), (95, 96)...(3, 4), (3, 2), (2, 2)$, where $(2, 2)$ is a PNE. In fact, BRD converge to $(2, 2)$ no matter where we start, and $(2, 2)$ is the only PNE!

Now, how would you play in this game? Experimental results [BCN05] show that most people play above 95, and the "winning" strategy (i.e., the one making the most money in pairwise comparisons) is to play 97 (which leads to an average payoff of 85). One potential explanation to what is going on is that people view the $2 penalty/reward as "too small" to start "undercutting"; indeed, other experiments [CGGH99] have shown that if we increase the penalty/reward, then people start declaring lower amounts, and after playing the game a certain number of times, converge on $(2, 2)$. So, in a sense, once there is enough money at play, best-response dynamics seem to be kicking in.

1.7 Mixed-strategy Nash Equilibrium

As mentioned, not all games have a PNE. We may also consider a generalized notion of a Nash equilibrium—referred to as a *mixed-strategy Nash equilibrium*—where, rather than choosing a single action, players choose a **mixed strategy**—that is, a *probability distribution* over actions: in other words, their strategy is to randomize over actions according to some particular probability distribution. The notion of a **mixed-strategy Nash equilibrium** is then defined to be a profile of (mixed strategies) with the property that no player can improve their *expected utility* by switching to a different strategy (See Section A.5 for preliminaries on expectations of random variables, as well as a discussion of expected utility.)

John Nash's theorem shows that every game with a finite action space and finite number of players has a mixed-strategy NE (even if it may not have a PNE). On a very high level, this is proved by relying on the observation that a Nash equilibrium is a fixed point to the best-response operator, and next applying a *fixed-point theorem* due to Kakutani [Kak41] which specifies conditions on the space X of inputs x and functions f for which a fixed-point $x \in f(x)$ exists: roughly, the requirements are that X is a "nice" subset (technically, "nice" means compact and convex) of vectors over real numbers, R^n, and that f satisfies an appropriate notion of continuity over X (one needs to generalize the standard notion of continuity to apply to set-valued functions since BR does not necessarily output a unique strategy). These conditions are not satisfied when letting X be the finite space of pure strategies, but are when enlarging X to become the space of all mixed strategies. For instance, as noted above, the Rock-Paper-Scissor's game does not have a PNE, but there is a mixed-strategy NE where both players uniformly randomize with probability $1/3$ over each of R, P, S.

Let us mention, however, that "perfectly" randomizing according to a mixed-strategy distribution may not always be easy. For instance, in the case of Rock-Paper-Scissors, if it were trivial to truly randomize, there would not be such things as extremely skilled players and world championships for the game! If we add a small cost for randomizing, mixed strategy Nash equilibria are no longer guaranteed to exist—in fact, even in the Rock-Paper-Scissors game.

To see this, consider a model where players need to choose an *algorithm* for playing the Rock-Paper-Scissors; the algorithm, perhaps using randomness, decides what action to play in the game. Finally, the player utility is defined as their utility in the underlying Rock-Paper-Scissors game *minus* some cost related to how much randomness the algorithm used. As we now argue, if the cost for randomizing is strictly positive (but the cost for algorithms using no randomness is 0), there cannot exists a Nash equilibrium where some player is using an algorithm that is actually randomizing: Assume for contradiction that player i is randomizing in some Nash equilibrium. Note that no matter what (mixed) strategy the other player, $-i$, is using, player i's best responses should always be some fixed (i.e., deterministic) pure strategy: for instance, if player $-i$ is randomizing but putting more weight on R, player i should best-respond by picking P, and if $-i$ is randomizing uniformly over all actions, any pure strategy is a best response. In particular, i can never best-respond by using an algorithm that randomizes over multiple actions, since we have assumed that randomizing is costly!

Thus, if there is a mixed-strategy Nash equilibrium in this game, it is also

a PNE (since nobody is actually randomizing), but we have already noted that Rock-Paper-Scissors does not have any PNE. Thus, in this Rock-Paper-Scissors with costly randomization game, there is not even a mixed-strategy Nash equilibrium.

While for the remainder of this course we will stick to the notion of PNE, there are many real-life games where PNE do not exist, and thus mixed-strategy Nash equilibria currently are our main tool for understanding how such games are played (despite the above-mentioned problem with them), and have indeed proven very useful.

Notes

The study of Game Theory goes back to the work by von-Neumann [Neu28] and the book by von-Neuman and Morgernstern [vM47]. We have only scratched the surface of this field; we refer the reader to [OR94] for a more detailed exposition of the area. The notion of a Nash equilibrium was introduced in [Nas50b]; John C. Harsanyi, John F. Nash Jr. and Reinhard Selten received the Nobel prize in Economics for "pioneering analysis of equilibria in game-theory" in 1994.

The Prisoner's Dilemma game was first considered by Merrill Flood and Melvin Dresher in 1950 as part of the Rand Corporation's investigations into game theory (Rand pursued this research direction because of the possible applications to global nuclear strategy). Albert Tucker formalized and presented this game in its current setting (in terms of the two prisoners); the name "the prisonner's dilemma" goes back to the work of Tucker.

The Traveller's dilemma was introduced by Basu [Bas94]. The existence of mixed-strategy Nash equilibrium was shown by Nash in [Nas50b]. The impossibility of mixed-strategy Nash equilibrium in games with costly randomization is due to Halpern and Pass [HP10].

Chapter 2

Graphs and Applications

Graphs are extremely important mathematical objects that arise quite often in real-life applications. For instance:

- In a *computer network*, we can model how all the computers are connected to each other as a graph. The nodes are the computers, and edges exist between computers that are connected to each other. This graph is obviously important for routing messages between the computers (to e.g., answer questions such as "What is the fastest route between two computers?").

- Consider a graph representing a map, where the edges are roads and the nodes are points of intersection and cities. How can we find the shortest path from point A to point B? (Some roads may be one-way, and we thus need the concept of *directed* edges to capture this.)

- We can model the Internet as a graph: nodes are webpages, and directed edges exist between nodes for which there exists a link from one webpage to the other. This structure is very important for Internet search engines: The relevance of a webpage is determined by how many links are pointing to it (and recursively how important those webpages are).

- *Social networks* can be modeled as graphs: each node could represent a person, with edges between the nodes representing people who are friends with one another.

In this chapter, we will discuss some basic properties of graphs, and present some simple algorithms for exploring them. (We return to some more advanced algorithms in Chapter 6.) We additionally present some initial connections

(a) A directed graph. (b) An undirected graph.

Figure 2.1: A basic example showing a directed and an undirected graph.

between graphs and game theory and show how to use graph representations to analyze games (and in particular BRD).

2.1 Basic definitions

Definition 2.1. A **directed graph** G is a pair (V, E) where V is a set of vertices (or nodes), and $E \subseteq V \times V$ is a set of edges.

Notice that directed graphs can contain *self-loops* $(v, v) \in E$; for instance, I can link to my own webpage. In directed graphs, the order of the nodes in an edge matters: if $u \neq v$, then $(u, v) \neq (v, u)$. But we can also define an *undirected* graph where order of nodes in an edge is irrelevant:

Definition 2.2. An **undirected graph** G is a pair (V, E) where V is a set of vertices (or nodes), and E is a set of sets $\{v, v'\}$ where $v, v' \in V$.

We often choose to simply represent an undirected graph as a directed graph where $(v', v) \in E$ if and only if $(v, v') \in E$ (i.e., we have directed edges in both directions).

Definition 2.3. A **path** or a **walk** in a graph $G = (V, E)$ is a sequence of vertices $(v_1, v_2, \ldots, v_k)$ such that there exists an edge between any two consecutive vertices, i.e. $(v_i, v_{i+1}) \in E$ for $1 \leq i < k$. A **cycle** is a path where $k \geq 1$ and $v_1 = v_k$ (i.e., it starts and ends at the same vertex). The length of the walk, path or cycle is k (i.e., the number of edges). We say that a node v' is **reachable** from v if there exists a path from v to v'.

A graph without cycles is called **acyclic**. A directed graph without cycles is called a **DAG** (a directed acyclic graph).

The degree of a vertex corresponds to the number of edges going out of, or coming into, a vertex. This is defined slightly differently for directed and undirected graphs.

Definition 2.4. In a directed graph $G = (V, E)$, the **in-degree** of a vertex $v \in V$ is the number of edges $e \in E$ coming into it (i.e., edges of the form $e = (u, v)$); the **out-degree** of v is the number of edges $e \in E$ going out of it (i.e., edges of the form $e = (v, u)$). The **degree** of v is the sum of the in-degree and the out-degree of v.

Definition 2.5. In an undirected graph $G = (V, E)$, the **degree** of $v \in V$ is the number of edges $e \in E$ going out of the vertex (i.e., of the form $e = u, v$), with the exception that self loops $e = v, v$ are counted twice.

This seemingly cumbersome definition of degree in undirected makes a lot of sense pictorially: the degree of a vertex corresponds to the number of "lines" going in/out from the vertex (and hence self loops in undirected graphs are counted twice).

The definition also leads to the following simple theorem and its corollary. (Neither the theorem nor the corollary will be essential for the sequel of the course, but they are interesting in their own right and serve as a nice illustration of the graph-theoretic concepts we have seen so far.)

Theorem 2.6. *Given a (directed or undirected) graph $G = (V, E)$, $2|E| = \sum_{v \in V} \deg(v)$.*

Proof. In a directed graph, each edge contributes once to the in-degree of some vertex and the out-degree of some, possibly the same, vertex. In an undirected graph, each non-looping edge contributes once to the degree of exactly two vertices, and each self-loop contributes twice to the degree of one vertex. In both cases we conclude that $2|E| = \sum_{v \in V} \deg(v)$. ■

An interesting consequence of this result is the following corollary:

Corollary 2.7. *In any (directed or undirected) graph G, the number of vertices with an odd degree is even.*

Proof. Let A be the set of vertices of even degree, and $B = V \setminus A$ be the set of vertices of odd degree. Then, by Theorem 2.6,

$$2|E| = \sum_{v \in A} \deg(v) + \sum_{v \in B} \deg(v)$$

Since the LHS and the first term of RHS is even, we have that $\sum_{v \in B} \deg(v)$ is even. In order for a sum of odd numbers to be even, there must be a even number of terms. ■

Although the statement of the corollary seems a bit obscure, it has a very natural intepretation in the context of social networks: it says that in *any* social network, the number of people with an odd number of friends is even. (This result is typically called the "handshake lemma" as a different interpretation of the result is that at any party, we have that the number of people that shake hands with an odd number of people is even.)

2.2 Connectivity

We turn to consider the notion of *connectivity*.

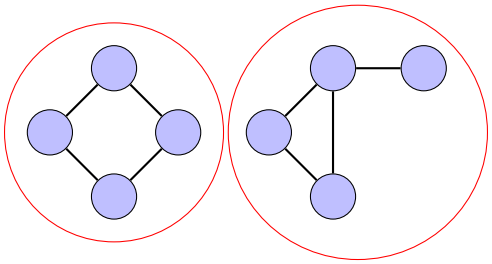
Definition 2.8. An undirected graph $G = (V, E)$ is **connected** if there exists a path between any two nodes $u, v \in V$ (note that a graph containing a single node v is considered connected via the length 0 path (v)). A directed graph $G = (V, E)$ is **connected** if the underlying undirected graph (i.e., the graph $G' = (V, E')$ where each edge $(u', v') \in E$ in G induces an undirected edge $(u', v'), (v', u') \in E'$ in G') is connected.[1] When a (directed or undirected) graph is not connected, we say that it is **disconnected**.

When a graph is disconnected, we may decompose the graph into smaller connected components.

Definition 2.9. Given a graph $G = (V, E)$, a **subgraph** of G is simply a graph $G' = (V', E')$ with $V' \subseteq V$ and $E' \subseteq (V' \times V') \cap E$; we denote subgraphs using $G' \subseteq G$. A **connected component** of an undirected graph $G = (V, E)$ is a *maximal* connected subgraph—that is, is a subgraph $H \subseteq G$ that is connected, and any larger subgraph H' (satisfying $H' \neq H$, $H \subseteq H' \subseteq G$) must be disconnected.

Let us note an intersting application of connectivity to social networks. It has been experimentally observed that in social networks (such as e.g., Facebook), although not everyone is in the same connected component, most people typically are: we informally refer to this component as the *giant component*. One reason for why most people are part of the same giant component is that if we had two (or more) large components in the graph, then all it takes to "merge" two large component into a single larger component is for one of the people in each component to become friends—it is very unlikely that this does not happen for *any* such pair of people. (This argument can

[1] We note that this notion of connectivity for directed graphs is typically referred to as "weak connectivity" in contrast to a notion of a "strong connectivity" which requires the existence of a (directed) path between any two nodes in the actual directed graph.



(a) Connected components of the graph are circled in red. Note that there are no edges between connected components.

Figure 2.2: Example of connected components.

be formalized in Erdös and Reyni's "random graph" model [ER59, Bol84]; we discuss the random graph model in more detail below.)

2.3 BFS and Shortest Paths

So how do we check if a graph is connected? The obvious way to do this is to start at some node v and check if we can reach every other node $v' \in V$ by traversing edges starting from there.

How do we check if there is a path from v to v'? The best way to do this is with **breadth-first search** (BFS). Breadth first search is a basic graph search algorithm that given a graph G and a starting node v proceeds as follows: visit all the neighbors v' of v, then visit all the currently unvisited neigbors of those nodes v', and so on and so forth. More precisely,

- Step 0: Mark v as visited.
- Step 1: Traverse the edge between v and any currently unvisited neighbor v' of v. Mark all those neighbors v' as visited (and for each such neighbor v', store a "back-pointer" to v).
- Step $k \geq 1$: For any node v' marked as visited in step $k-1$, traverse any edge between v' and some currently unvisited neigbor v'' of v'; mark all those neighbors v'' as visited (and for each such neighbor v'', store a "back-pointer" to v').

See Figure 2.3 for an illustration of the execution of the BFS algorithm.

As we now show, the BSF algorithm can be used not only to determine whether there *exists* a path from v to v', but also to find some *shortest path* (i.e., a path of minimal length) between v and v'.

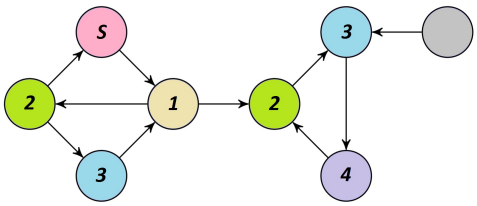


Figure 2.3: An illustration of breadth-first search. Starting from node S, in the first step we visit all of the neighbors of S (marked with a 1), in the second step we visit all of the unvisited neighbors (marked with a 2) of each node visited in the first step, and so on. Notice that, in four steps, BFS visits every node except for the top-right node (unlabeled), which clearly has no path from S.

Claim 2.10. *For all vertices v, v', if there exists a path from v to v', the BFS algorithm starting from vertex v traverses some shortest path from v to v' (and this shortest path can be retrieved by following the "back-pointers" from v').*

Proof. Assume for contradiction that there exist a path from v and v', but that the BFS algorithm does not traverse *any* shortest path from v to v'. Let $(v_0 = v, v_1, v_2, \ldots, v_n = v')$ be a shortest path from v and v', and let v_i be the *first* node on the path such that BFS does not traverse a path of length (at most) i from v to v_i. Such a node must exists because, by assumption, BFS does not traverse a path of length n from $v = v_0$ and $v' = v_n$; additionally $i \geq 1$, since $v_0 = v$ is the starting point.

Since v_i is the first node on the path for which BFS fails to traverse a path of length at most i, it must have traversed a path of length at most $i - 1$ to get to v_{i-1}. Consequently, after visiting v_{i-1}, BFS will visit v_i since it is a neighbor of v_{i-1} that still is unvisited at this point (or else BFS would have traverse a path of length at most $i - 1$ to get from v to v_i). Thus, BFS traverses a path of length at most i from v to v_i, which is a contradiction. ■

As a consequence of the above claim, and observing that BFS will only visit each node once (and thus its running time is polynomial in the number of nodes in the graphs), we have the following theorem.

Theorem 2.11. *Shortest paths in a graph G can be found in time polynomial in $|G|$.*

Since running BFS from a node v visits all the nodes that are reachable from v (by Claim 2.10), we also have the following result.

Theorem 2.12. *Given any node v in G, the set of nodes that are reachable from v in G can be found in time polynomial in $|G|$.*

We can also use BFS to find the connected components of a graph: Start at any node v and use BFS to find the connected component of v; mark all those nodes as component 1. Next, take some currently unmarked node v', and again use BFS to find the connected component of v'; mark all those notes as component 2. Continue in the same manner until all nodes have been marked.

Some applications of BFS and shortest paths to social networks include:

- The "Bacon number" of an actor or actress is the shortest path from an individual to the actor Kevin Bacon in the graph where the nodes are actors and actresses, and edges connect people who star together in a movie. The "Erdös number" is similarly defined to be the distance of an individual to the mathematician Paul Erdös in the co-authorship graph.

- Milgram's *small-world* experiment [Mil67] (a.k.a. *the six degrees of separation*) demonstrates that everyone is approximately 6 steps away from anyone else in a social network: random people in Omaha were asked to forward a letter to a "Mr. Jacobs" in Boston, but they could only forward the letter through someone they knew on a first-name basis. Surprisingly, the average path length was roughly 6. This "small-world phenomenon" has since been reconfirmed in multiple subsequent studies. In 1998, Watts and Strogatz [WS98] demonstrated a mathematical model that accounts for this phenomenon.

- Milgram's experiment shows that not only are people connected through short paths, but they also manage to efficiently route messages given only their *local knowledge* of the graph structure. Jon Kleinberg's [Kle00] *local search model* provides a refinement of the Watts-Strogatz small-world model which can explain how such routing is possible.

To get some intuition for why small world effects happen, let us consider Erdös and Reyni's "random graph" model [ER59]:

The random graph model. Consider a graph with n nodes, where every pair of nodes has an edge between them (determined independently and with no self-loops) with probability $\frac{1}{2}$. What is the probability that any two vertices v and v' have a path of length at most 2 between them?

Given any third vertex z, the probability of a path $\{(v,z),(z,v')\}$ is only $\frac{1}{2} * \frac{1}{2} = \frac{1}{4}$. However, since there are $n-2$ possible "third vertices" for this path, the probability that two nodes are *not* connected by any path of length exactly 2 is

$$\left(1 - \frac{1}{4}\right)^{n-2} = \left(\frac{3}{4}\right)^{n-2}.$$

By the Union Bound (see Corollary A.7 in Appendix A), we thus have that the probability that there exists *some* pair of nodes that is more than distance 2 apart is

$$\begin{aligned}
\Pr\left[\bigcup_{u \neq v} u,\, v \text{ has distance} > 2\right] &\leq \sum_{u \neq v} \Pr[u,\, v \text{ has distance} > 2] \\
&\leq \sum_{u \neq v} \Pr[u,\, v \text{ has distance} \neq 2] \\
&= \frac{n(n-1)}{2}\left(\frac{3}{4}\right)^{n-2}
\end{aligned}$$

This quantity decreases very quickly as the number of vertices, n, increases. Therefore, it is extremely likely that *every* pair of nodes is at most distance 2 apart.

Needless to say, in a real social network, people are far less likely than probability $\frac{1}{2}$ to be connected, but the number n of nodes is extremely large. Hence, similar arguments apply to show that the average separation between two nodes is relatively small.

Notes

We refer the reader to [CLRS09] and [KT05] for a more in-depth study of graph algorithms, and to [EK10] for connections between graphs properties and social networks.

Chapter 3

Analyzing Best-Response Dynamics

In this chapter, we present the first simple connection between game-theory and graph-theory: we use graphs to analyze games and best-response dynamics. Using this graph-theoretic approach, we will be able to give a simple characterization of the class of games for which best-response dynamics converge. Both the characterization, as well as the method used to prove it, will be useful in the sequel.

3.1 A Graph Representation of Games

We can use graphs to analyze (normal-form) games as follows. Given a game $\mathcal{G} = (n, A, u)$, consider a directed graph G where the set of vertices is the set A of action profiles, and where we put a (directed) edge between nodes $\vec{a}$ and $\vec{a}'$ if and only if there exists some player i such that a) $\vec{a}$ and $\vec{a}'$ differ only in component i (i.e., i's action) and b) $a_i \notin BR_i(\vec{a})$ but $a'_i \in BR_i(\vec{a})$; that is, draw an edge between action profiles if we can go from the first action profile to the second in one step of BRD (i.e. one player improving its utility by switching to its best response). See Figure 3.1 and 3.2 for examples of such graphs for Bach-Stravinsky and the Traveler's Dilemma.

We can now reinterpret BRD and PNE using standard graph-theoretic notions:

- The BRD process can be interpreted as simply starting at any node $\vec{u}$ in G and then picking any outgoing edge, traversing it to reach a new

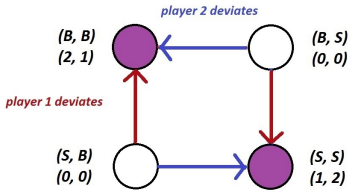


Figure 3.1: A BRD graph for the Bach-Stravinsky Game. Sinks (i.e., equilibria) are marked in purple. Observe that there are multiple sinks reachable from each non-equilibrium state, depending on which player deviates.

node $\vec{u}'$, picking any outgoing edge from it and so on. That is, we take a walk from any node in the graph.

- BRD *converges* in $\mathcal{G}$ if and only if there are no cycles in G—that is, G is a DAG. This means that we eventually reach a node that does not have any outgoing edges and thus the path ends.

- By Claim 1.9, $\vec{a}$ is a PNE if and only if $\vec{a}$ is a, so-called, *sink* in the graph G—that is, a node without any outgoing edges. If a node is a sink in G, nobody can improve their utility by best responding, and thus $\vec{a} \in BR(\vec{a})$.

Note that it is not necessary that all nodes reach the same sink for BRD to converge; as in the Bach-Stravinsky game, some games can have multiple equilibria, or even multiple equilibria reachable from the same starting profile.

3.2 Characterizing Convergence of BRD

We can now use this graph representation to characterize the set of games for which BRD converges. Given a game $\mathcal{G} = (n, A, u)$, we define a **potential** (or "energy") function $\Phi : A \to \mathbb{Z}$, which maps outcome profiles to integers (denoting the "energy level"). A particularly natural potential function (which we will use later) is the **utilitarian social welfare**, or simply **social welfare** (SW), defined as the sum of all players' utilities:

$$\Phi(a) = \text{SW}(a) = \sum_{i \in n} u_i(a)$$

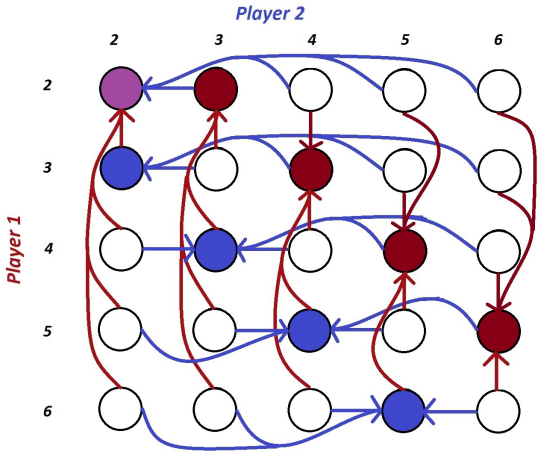


Figure 3.2: A small part of the BRD graph for the Traveler's Dilemma, illustrating why this game converges to the equilibrium of (2, 2). Nodes that can be reached as a best-reponse by player 1 are marked in red, and those than can be reaches as a best reponse by player 2 are marked in blue; the sink is marked in purple.

The notion of an *ordinal* potential function will be useful. Roughly speaking, a potential function is said to be *ordinal* if any profitable single-player deviation increases the potential:

Definition 3.1. $\Phi : A \to \mathbb{Z}$ is an **ordinal potential function** for a game $\mathcal{G} = (n, A, u)$ if, for every action profile $a \in A$, every player i, and every action $a'_i \in A_i$, if

$$u_i(a'_i, a_{-i}) > u_i(a_i, a_{-i})$$

then

$$\Phi(a'_i, a_{-i}) > \Phi(a_i, a_{-i})$$

We may also consider a weaker form of a potential function, which we refer to as *weakly ordinal*, which simply requires the potential to increase for any profitable single-player deviation *to a best response*.

Definition 3.2. $\Phi : A \to \mathbb{Z}$ is a **weakly ordinal potential function** for a game $\mathcal{G} = (n, A, u)$ if, for every action profile $a \in A$, every player i, and every action $a'_i \in A_i$ such that $a_i \notin BR_i(\vec{a})$ but $a'_i \in BR_i(\vec{a})$, we have that

$$\Phi(a'_i, a_{-i}) > \Phi(a_i, a_{-i})$$

Note that if Φ is ordinal, then it is clearly also weakly ordinal, since if $a_i \notin BR_i(\vec{a})$ but $a_i' \in BR_i(\vec{a})$ then $u_i(a_i', a_{-i}) > u_i(a_i, a_{-i})$.

We now have the following characterization of the class of games for which BRD converges.

Theorem 3.3. *BRD converges in $\mathcal{G}$ if and only if $\mathcal{G}$ has a weakly ordinal potential function.*

Proof. We will prove each direction separately.

Claim 3.4. *If $\mathcal{G}$ has a weakly ordinal potential function, then BRD converges in $\mathcal{G}$.*

Proof. Consider an arbitrary starting action profile (node) $\vec{a}$, and consider running BRD starting from $\vec{a}$. Each time some player best-responds, the potential Φ increases. Since the game has a finite number of states, there is only a finite number of possible potentials; thus, after a finite number of best-response steps, we must have reached the highest possible potential, which guarantees convergence as no more "deviations" are possible from that point. ■

Claim 3.5. *If BRD converges in $\mathcal{G}$, then $\mathcal{G}$ has a weakly ordinal potential function.*

Proof. Consider some game $\mathcal{G}$ where BRD converges, and consider the induced graph representation G of $\mathcal{G}$. As noted above, since BRD always converge, G is a DAG, and thus every path from any starting point $\vec{a}$ must eventually lead to a sink. We construct a potential function Φ for $\mathcal{G}$ by for each action profile $\vec{a}$, letting $\Phi(\vec{a}) = -\ell(\vec{a})$ where $\ell(\vec{a})$ is the length of the *longest* path from $\vec{a}$ to *any* sink s in G.

We now show that each time we traverse an edge in the BRD graph G (i.e., take one step in the BRD process) the potential increases—that is, ℓ decreases. Assume we traversed the edge $(\vec{a}, \vec{a}')$ and that $\ell(\vec{a}') = k$. Let us then argue that $\ell(\vec{a}) \geq k + 1$. This directly follows since there exists some path p of length k from $\vec{a}'$ to some sink s, and thus there exists a path of length $k + 1$ from $\vec{a}$ to s by simply first traversing $(\vec{a}, \vec{a}')$ and then following p. ■

The theorem directly follows from the above two claims. ■

In the sequel, we will use Theorem 3.3 as a way to show that BRD converges in the games we will be considering. In particular, for all those games, we will exhibit an ordinal potential function (which, as noted above, must also

be weakly ordinal), and thus Theorem 3.3 can be used to deduce that BRD converges.

3.3 Better-Response Dynamics

The fact that a game has an ordinal (as opposed to just weakly ordinal) potential function is interesting in its own right. In such games, a more general type of best-response dynamics—referred to as **better-response dynamics**—will always converge: better-response dynamics proceeds exactly as best-response dynamics except that players can switch to *any profitable deviation* (as opposed to just the best one). More precisely, at every step in the dynamics:

- Pick *any* player i for which $a_i \notin BR_i(\vec{a})$, and replace a_i by *any* action in $a' \in A_i$ such that
$$u_i(a'_i, a_{-i}) > u(\vec{a})$$

In particular, a'_i may not necessarily be the optimal response (i.e., a best-response) to a_{-i}. For instance, consider the outcome $(70, 50)$ in the Traveler's Dilemma: player 1 is currently getting $50 - 2 - 48$ in utility, and could "better-respond" by choosing $50, 49, 48, 47$; in contrast, the only best-response is choosing 49 (which would yield utility 51). See Figure 3.3 for an illustration of the (very complicated) "better-response dynamics graph" for the Traveler's Dilemma.

Better-response dynamics captures situations where it may be hard for players to always find some best-response (for instance, if the action space is large) but the players clearly want to change whenever they find some action that improves their utility—i.e., the players simply "myopically" try to improve their utility. Note that, since any valid execution of BRD is also a valid execution of the better-response dynamics, whenever better-response dynamics converges, so does BRD. More generally, we now have the following characterization of the class of games for which better-response dynamics converges.

Theorem 3.6. *Better-response dynamics converges in $\mathcal{G}$ if and only if $\mathcal{G}$ has an ordinal potential function.*

Proof. The proof is essentially identical to the proof of Theorem 3.3, except that in the "only-if direction" we replace the best-response graph by a "better-response graph". ■

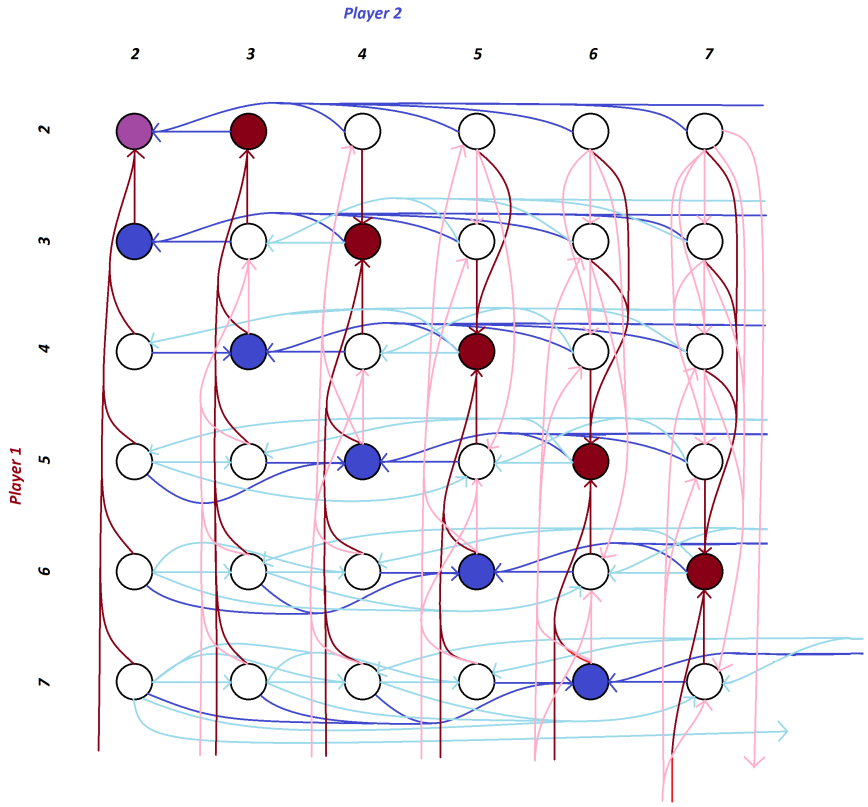


Figure 3.3: A small part of the better-response dynamics graph for the Traveler's Dilemma. The edges that are present in better-response dynamics but not in BRD are indicated in lighter colors. Nodes that can be reached as a best-reponse are colored.

As mentioned, in the sequel, we will use ordinal potential functions to demonstrate convergence of BRD; those proofs thus also show convergence of better-response dynamics.

3.4 Games without PNE

As mentioned in Chapter 1, there are games—such as the Rock-Paper-Scissors game—without any PNE. For such games, by Claim 1.9, BRD cannot converge. We end this section by illustrating in Figure 3.4 the BRD graph for the Rock-Paper-Scissors game.

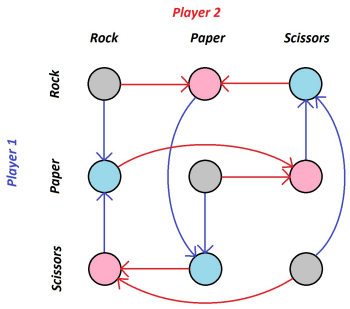


Figure 3.4: A BRD graph for a game with no pure-strategy Nash equilibrium (rock-paper-scissors); notice that there is no "sink" to which the process of best-response dynamics converges.

Notes

The notion of an ordinal potential function and the characterization of games for which better-response dynamics converge is due to Monderer and Shapley [MS96]. The use of potential functions to prove the existence of PNE dates back to the work by Rosenthal [Ros73].

Chapter 4

Coordination in Social Networks

Recall the iPhone/Android game we considered in the introduction. We now have the necessary tools for analyzing it. We begin by considering a simple form of this game on a social network described by an undirected graph.

4.1 Plain Networked Coordination Games

Given some *undirected* graph G, consider a game $\mathcal{G}$ being played by the nodes of the graph (i.e., the nodes represent the players): each node on the graph selects a *single* action (a "product")—either X or Y—and participates in the following coordination game (where both players get the same utility in every outcome) with *each* of its neighbors (assume $x, y > 0$):

1 \ 2	$(*, X)$	$(*, Y)$
$(X, *)$	(x, x)	$(0,\ 0)$
$(Y, *)$	$(0,\ 0)$	(y, y)

That is,

- If they "match", they get the same positive utility $Q(X, X) = x$ or $Q(Y, Y) = y$;
- If they "mismatch", they get utility $Q(X, Y) = Q(Y, Y) = 0$;

Note that the utility, x, of matching on X may be different from the utility, y, of matching on Y. Without loss of generality, we assume $x \geq y$; that is,

product X is *at least as good* as product Y. The utility of a node is then finally defined to be the *sum of its utilities* in all the coordination games it participates in. Formally,

Definition 4.1. Given an undirected graph $G = (V, E)$, we say that the game $\mathcal{G} = (n, A, u)$ is **plain networked coordination game induced by G and the coordination utility** $Q : \{X, Y\}^2 \to \mathbb{N}$ if:

- $V = [n]$, $Q(X, Y) = Q(Y, X) = 0$, $Q(X, X) \geq Q(Y, Y) \geq 0$.
- $A = \{X, Y\}^n$
- $u_i(\vec{a}) = \sum_{j \in N(i)} Q(a_i, a_j)$, where $N(i)$ is the set of neighbors of the node i in G.

In the sequel, to simplify notation, we will have some particular plain networked coordination game in mind, and we will let x denote the utility of coordinating at X (i.e., $Q(X, X)$) and y the utility of coordinating at Y (i.e., $Q(Y, Y)$).

A Simple Decision Rule: The Adoption Threshold Consider a node v that has d neighbors (friends), a fraction p of whom choose X and the remaining fraction $(1-p)$ of whom choose Y. Then v's utility for choosing X is pdx and its utility for choosing Y is $(1-p)dy$. What action should v take to maximize its utility (i.e., to best respond to the actions of the other players)?

- Choose X if $pdx > (1-p)dy$—that is, $p > \frac{y}{x+y}$.
- Choose Y if $pdx < (1-p)dy$—that is, $p < \frac{y}{x+y}$.
- Choose either X or Y if $p = \frac{y}{x+y}$.

So, no matter what the *number* of v's neighbors is, the decision ultimately only depends on the *fraction* of its neighbors choosing X or Y (and the game utilities). We refer to the ratio $t = \frac{y}{x+y}$ as the **(global) adoption threshold** for product X.

4.2 Convergence of BRD

We now turn to the question of analyzing what equilibria look like in these games. Clearly, everyone choosing either X or Y is a PNE (X is the "better" equilibrium, but Y is still a PNE). But there are other equilibria as well, such as the ones illustrated in Figure 4.1.

Figure 4.1: Illustrations of equilibria in a plain networked coordination game. The left example is an equilibrium when the adoption threshold is $\frac{1}{2}$ or more; the right example shows how to sustain a "non-trivial" equilibrium even when the adoption threshold is smaller, namely, $\frac{1}{3}$ (or more).

Intuitively, the reason why these "non-trivial" equilibria arise is that there is some network structure that "blocks" the better action X from spreading to more nodes in the network. We will return to an analysis of this phenomenon in Chapter 5; for now, we will consider the question of whether we can converge to an equilibrium using best-response dynamics.

As we observed in the previous chapter, BRD in a game $\mathcal{G}$ converges if and only if $\mathcal{G}$ has an ordinal potential function. Here, given a plain coordination game $\mathcal{G} = (n, A, u)$, we will consider the *social welfare* function given by

$$\Phi^{\mathcal{G}}(\vec{a}) = \mathrm{SW}^{\mathcal{G}}(\vec{a}) = \sum_{i \in n} u_i(\vec{a})$$

Whenever the game $\mathcal{G}$ is clear from context, we omit it as a superscript for Φ and SW. Notice that by expanding out the definition of u, we get

$$\Phi(\vec{a}) = \sum_{(i,j) \in E} Q(a_i, a_j) = 2 \sum_{\{i,j\} \in E} Q(a_i, a_j) \tag{4.1}$$

Let us now prove that Φ is an ordinal potential function of the game.

Claim 4.2. *$\Phi^{\mathcal{G}}$ is an ordinal potential function for any plain networked coordination game $\mathcal{G}$.*

Proof. Consider an action profile $\vec{a}$ and some player i who can improve their utility by deviating to a_i'; that is,

$$u_i(a_i', a_{-i}) > u_i(a_i, a_{-i}) \tag{4.2}$$

We need to show that $\Phi(a_i', a_{-i}) > \Phi(a_i, a_{-i})$, or equivalently that $\Phi(a_i', a_{-i}) - \Phi(a_i, a_{-i}) > 0$. Notice that when considering $\Phi(a_i', a_{-i}) - \Phi(a_i, a_{-i})$, the only

games that are affected are those between i and the neighbors of i, $N(i)$. So, by Equation 4.1, the difference in the potential is

$$2 \sum_{j \in N(i)} (Q(a_i', a_j) - Q(a_i, a_j)) = 2(u_i(a_i', a_{-i}) - u_i(a_i, a_{-i}))$$

which by Equation 4.2 is strictly positive. ■

Thus, by Theorem 3.3 (which proved that BRD converges if and only if the game has an ordinal potential function), we directly get the following theorem.

Theorem 4.3. *BRD converges in every plain networked coordination game.*

Socially optimal outcomes We say that an outcome $\vec{a}$ is **socially optimal** if the outcome that maximizes social welfare; that is, the social welfare in the outcome $\vec{a}$ is at least as high as the social welfare in any other outcome $\vec{a}'$. In other words, $\vec{a}$ is socially optimal in $\mathcal{G}$ if

$$\vec{a} \in \arg\max_{\vec{a} \in A} SW^{\mathcal{G}}(\vec{a}).$$

Note that there may not necessarily exists a unique outcome that maximizes social welfare, so several outcomes may be socially optimal.

Let us also remark that if social welfare is an ordinal potential function (as we showed was the case for plain networked coordination games), then whatever outcome we start off it, if people deviate to make themselves better off, they also make the outcome better for "the world in aggregate" (although some player can get worse off). In particular, this implies that if we start out with a socially optimal outcome, BRD will stay there. Note that this is not always the case: for instance in the Prisoner's Dilemma, the socially optimal outcome is for both players to cooperate, but BRD moves away from this outomes (as it is not a PNE).

The fact that this happens here, however, should not be too surprising: if $x > y$, there is a unique outcome that maximizes SW (namely, everyone choosing X) and this outcome is a PNE, thus BRD will stay there; if $x = y$, there are two outcomes (everyone choosing X or everyone choosing Y) which maximize SW, both of which are PNE.

4.3 Incorporating Intrinsic Values

So far we have assumed that players' utilities depend only on their coordination utilities; in particular, this implies that everyone has the same "intrinsic

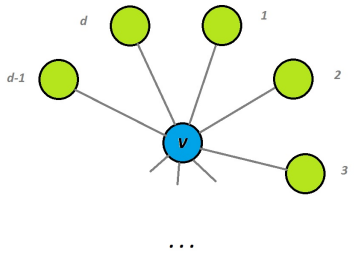


Figure 4.2: The star graph described in the example, with the equilibrium state highlighted (X in blue, Y in green).

value" for each product. In reality, people may have different intrinsic value for different products (e.g. in the absence of network effects, some people naturally prefer iPhones, and others Androids).

To model this, we can consider a more general class of games where utility is defined as follows:

$$u_i(\vec{a}) = R_i(a_i) + \sum_{j \in N(i)} Q(a_i, a_j) \tag{4.3}$$

Here, $R_i(a_i)$ denotes the intrinsic value of a_i to player i. Formally, the notion of a **networked coordination game** is defined just as in Definition 4.1, except that we now also parametrize the game by an **intrinsic value function** $R_i : \{X, Y\} \to \mathbb{N}$ for each node i and use Equation 4.3 to define utility.

Notice that nodes can no longer employ the *same* simple decision rule of checking whether the fraction of its neighbors playing X exceeds some fixed *global* threshold t—rather, each node has its own **subjective (or local) adoption threshold** $t(i)$ which depends on the number of i's neighbors and its intrinsic value function R_i (excercise: derive what the subjective adoption threshold is). Also, notice that it is now no longer clear what equilibria look like, or even that they exist, since there might exist a conflict between the intrinsic value of an action to a player and the desire to coordinate with their friends!

Example. Consider the simple star graph in Figure 4.2 with $d + 1$ nodes, consisting of a "central" node v connected to d neighbors.

- Let $R_v(X) = d + \epsilon, R_v(Y) = 0$.
- For all other nodes v', $R_{v'}(X) = 0$ and $R_{v'}(Y) = 1 + \epsilon$.

- Let $Q(X, X) = Q(Y, Y) = 1$ (i.e. $x = y = 1$ in the game as described above).

So every node's coordination game indicates no network preference between X and Y. But v intrinsically "strongly" prefers X, while all other nodes intrinsically "weakly" prefer Y.
What happens if everyone chooses X in this game?

- v has maximized their intrinsic utility ($d+\epsilon$) and maximized coordination utility (1 on each of d edges, for a total of d), so there is no incentive to switch.
- Other nodes currently have no intrinsic utility and 1 in coordination utility, so it benefits them to switch to Y instead, receiving $1 + \epsilon$ in intrinsic utility and losing 1 in coordination utility.
- This state has social welfare $3d + \epsilon$, but is not an equilibrium (since nodes $v' \neq v$ gain ϵ by switching actions).

What happens if everyone chooses Y?

- Neighbors of v have maximized their intrinsic utility ($1 + \epsilon$) and maximized coordination utility (1), so there is no incentive for them to switch.
- v currently has no intrinsic utility and d in coordination utility (1 on each edge), so it benefits v to switch to X instead, receiving $d + \epsilon$ in intrinsic utility and losing d in coordination utility.
- This state has social welfare $3d + d\epsilon$, but also is not an equilibrium.

In fact, if we apply the BRD process we will *always* up end in a state where v chooses X and the other nodes choose Y, no matter where we start. This follows from the fact that X is strictly dominant for v and Y is strictly dominant for all the other nodes; thus v playing X and everyone else playing Y is the only PNE (and by Claim 1.10, BRD quickly converges to it).

But in this state, there is no "coordination" utility; v receives $d + \epsilon$ in intrinsic utility and the other nodes receive $1 + \epsilon$ each, for a total of only $2d + (d + 1)\epsilon$ in social welfare. There is actually quite a significant gap—a factor of close to $\frac{3}{2}$—between the equilibrium and the "coordination" states where everyone plays either X or Y (in fact, it is not hard to see that the state where everyone plays Y is the only socially optimal outcome in this game; we leave it as an exercise to prove this). We refer to the existence of such a gap as an **inefficiency of equilibria** (i.e., that selfish behavior leads to "inefficient" outcomes).

Furthermore, notice that this gap is essentially independent of the number of nodes, $d + 1$, as long as $d \geq 1$. Thus, an even simpler example illustrating

this gap is obtained by simply considering the case when $d = 1$—that is, we are simply playing a coordination game between two players. (We consider the slightly more complicated star example as it illustrates the issue even in a connected graph with a large number of nodes.)

Can we still prove that equilibria always exist in these games by showing that best-response dynamics converge? Social welfare is no longer an ordinal potential function; as a counterexample, consider the state in the graph above where all nodes choose Y. If v switches to X, improving its own utility, the social welfare actually *decreases* by $d - \epsilon$ (from $3d + d\epsilon$ to $2d + (d+1)\epsilon$), even though v actually only increases its utility by ϵ!

However, we can choose a better potential function which is ordinal; namely, let

$$\Phi'(\vec{a}) = \sum_{\{i,j\} \in E} Q(a_i, a_j) + \sum_{i \in V} R_i(a_i)$$

That is, we sum coordination utilities over every edge *once* (rather than twice as in the definition of social welfare; see Equation 4.1), and add the intrinsic values for each node.

Theorem 4.4. *BRD converges in every Networked Coordination Game.*

Proof. Let Φ' be above-described potential function; by Theorem 3.3, it suffices to show that it is ordinal. Consider an action profile $\vec{a}$ and some player i who can improve their utility by deviating to a_i'; that is,

$$u_i(a_i', a_{-i}) > u_i(a_i, a_{-i})$$

We will show that $\Phi'(a_i', a_{-i}) - \Phi'(a_i, a_{-i}) > 0$. Once again, in considering $\Phi'(a_i', a_{-i}) - \Phi'(a_i, a_{-i})$, note that only coordination games between i and $N(i)$ and the intrinsic value for i are affected by changing a_i to a_i'. So,

$$\Phi'(a_i', a_{-i}) - \Phi'(a_i, a_{-i}) = \sum_{j \in N(i)} (Q(a_i', a_j) - Q(a_i, a_j)) + (R_i(a_i') - R_i(a_i))$$

$$= u_i(a_i', a_{-i}) - u_i(a_i, a_{-i}) > 0$$

■

4.4 The Price of Stability

As we saw in the earlier example, BRD might decrease social welfare—in particular, even a single player best-responding *once* can significantly bring

down the social welfare. A natural question thus arising is: How bad can the "gap" between the social welfare in the best equilibrium and in the socially-optimal outcome be? We denote by

$$MSW^{\mathcal{G}} = \max_{\vec{a} \in A} SW^{\mathcal{G}}(\vec{a})$$

the **maximum social welfare (MSW)** obtainable in the game $\mathcal{G}$. In games with non-negative utilities (such as networked coordination games), we refer to the ratio between the MSW and the SW of the *best* PNE as the *Price of Stability* in the game.

Definition 4.5. The **Price of Stability** in a game $\mathcal{G}$ with non-negative utilities is defined as

$$\frac{MSW^{\mathcal{G}}}{\max_{\vec{a} \in PNE^{\mathcal{G}}} SW^{\mathcal{G}}(\vec{a})}$$

where $PNE^{\mathcal{G}}$ denotes the set of PNE in $\mathcal{G}$.

Roughly speaking, the price of stability measures by how much selfish behavior *necessarily* degrades the "efficiency" of outcomes. (There is also a related notion of *Price of Anarchy* which is defined as the ratio between MSW and the *worst* PNE—this notion instead measure by how much selfish behavior can, in the most pessimistic case, degrade performance. In the sequel, however, we will focus only on studying the Price of Stability).

As we now show, selfish behavior cannot degrade performance by more than a factor 2.

Theorem 4.6. *In every Networked Coordination Game $\mathcal{G}$, there exists a PNE $\vec{a}'$ such that*

$$SW^{\mathcal{G}}(\vec{a}') \geq \frac{1}{2} MSW^{\mathcal{G}}$$

(In other words, the Price of Stability in $\mathcal{G}$ is at most 2.)

Proof. Observe that for the potential function Φ' defined above, for every $\vec{a} \in A$,

$$SW(\vec{a}) \geq \Phi'(\vec{a}) \geq \frac{SW(\vec{a})}{2}$$

Now, pick any outcome $\vec{a}$ that maximizes $\mathrm{SW}(\vec{a})$ (and thus achieves a SW of MSW). Then run BRD starting from $\vec{a}$ until it converges to some outcome profile $\vec{a}'$—as shown in Theorem 4.4 it will always converge, and this final

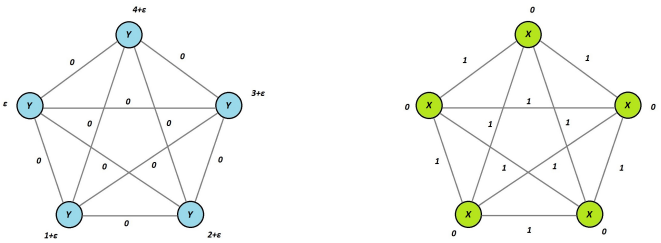


Figure 4.3: The construction presented in Theorem 4.7 for $n = 5$, detailing the intrinsic utilities (next to the nodes) and coordination utilities (on the edges) achieved in (left) the equilibrium state and (right) the maximum achievable social welfare.

outcome is a PNE. While SW may decrease at each step, as shown in the proof of Theorem 4.4 Φ' can only increase. Hence, we have

$$SW(\vec{a}') \geq \Phi'(\vec{a}') \geq \Phi'(\vec{a}) \geq \frac{SW(\vec{a})}{2} = \frac{MSW}{2}$$

as desired. ■

Recall that our "star graph" exhibited an example of a game with a gap of close to $\frac{3}{2}$ between the best equilibrium and the socially optimal outcome, whereas Theorem 4.6 shows that the gap can never be more than 2. It turns out that the gap of 2 is actually tight, but showing this requires a slightly more elaborate analysis appealing to iterated strict dominance.

Theorem 4.7. *For every $n \in \mathbb{N}, \epsilon > 0$, there exists some n-player Networked Coordination Game $\mathcal{G}$ such that for every PNE $\vec{a}'$ in $\mathcal{G}$ we have that*

$$SW^{\mathcal{G}}(\vec{a}') \leq (\frac{1}{2} + \epsilon) MSW^{\mathcal{G}}$$

(In other words, the Price of Stability can be arbitrarily close to 2.)

Proof. Let $G = ([n], E)$ be a complete graph on n nodes. Let $Q(X, X) = 1, Q(Y, Y) = 0$ and for every player i, let $R_i(X) = 0, R_i(Y) = n - i + \epsilon$. (See Figure 4.3.) Let $\vec{a} = (X, X, \ldots, X)$; note that $SW(\vec{a}) = 2|E| = n(n-1)$, and so $MSW \geq n(n-1)$. Let us next analyze the PNE in this game. We will show that $\vec{b} = (Y, Y, \ldots, Y)$ is the only strategy profile that survives iterated strict dominance (ISD); by Claim 1.8 (showing that if a unique strategy survives ISD, it must be the unique NE of the game), it thus follows that $\vec{b}$ is also the only PNE of the game.

Showing that only $\vec{b}$ survives ISD We will show by induction player after the kth round of ISD, player k must have removed action X. The base case ($k = 0$) is trivial. For the induction step, assume the claim is true up to round k and let us show it for $k+1$. In the $(k+1)$th round, by the induction hypothesis, players $1, \ldots, k$ must be playing Y. Therefore, player $k+1$ can get utility at most $n-k-1$ by playing X, whereas it can guarantee itself $n-k-1+\epsilon$ (in intrinsic value) by playing Y; thus X is strictly dominated and must be removed, which proves the induction step.

Concluding the Price of Stability Bound Note that

$$SW(\vec{b}) = \sum_{i=1}^{n} R_i(Y) = \sum_{i=1}^{n}(n-i+\epsilon) = \frac{n(n-1)}{2} + n\epsilon$$

Thus,

$$\frac{SW(\vec{b})}{MSW} \leq \frac{\frac{n(n-1)}{2} + n\epsilon}{n(n-1)} = \frac{1}{2} + \frac{\epsilon}{n-1}$$

which may be taken arbitrarily close to $1/2$ by choosing an appropriately small ϵ. ■

4.5 Incorporating Strength of Ties

We finally consider an even more general networked coordination model where we place a "weight" $w_{i,j} = w_{j,i}$ on each (undirected) edge $\{i, j\}$—think of this weight as the "strength of the friendship" (measured e.g., by how many minutes we spend on the phone with each other, or how many messages we send to each other)—and now also weigh the coordination utility of the game between i and j by $w_{i,j}$. That is,

$$u_i(\vec{a}) = R_i(a_i) + \sum_{j \in N(i)} w_{i,j} Q(a_i, a_j)$$

We note that the potential function, as well as social welfare, arguments made above (i.e. Theorem 4.4 and Theorem 4.6) directly extend to this more general model, by defining

$$\Phi'(\vec{a}) = \sum_{\{i,j\} \in E} w_{i,j} Q(a_i, a_j) + \sum_{i \in V} R_i(a_i).$$

Note, however, that a players decision whether to play X is no longer just a function of the fraction of its neighbors playing X; rather, we now need to

consider a **weighted subjective threshold** $t(\cdot)$ where node i switches to X whenever the fraction of its neighbors j *weighted* by $w(i,j)$ exceeds t.

Notes

The notion of a networked coordination games (the way we consider them) was introduced by Young [You02]. Young also showed the existence of PNE in such games.

The gap between the maximum social welfare and the social welfare in equilibria (i.e, the "inefficiency" of equilibria) was first studied by Koutsoupias and Papadimitriou [KP09], but the notion considered there—*"price of anarchy"*—considered the gap between MSW and the *worst* PNE. In contrast, the notion of *price of stability* (which we considered here) considers the gap between MSW and the *best* PNE. This notion was first studied in [SM03], and the term "price of stability" was first coined in [ADK$^+$08]; [ADK$^+$08] was also the first paper to study price of stability using potential functions. The results on price of stability in coordination games that we have presented here are from Morgan and Pass [MP16].

It turns out that the star graph example we presented (and its gap of $\frac{3}{2}$) is also tight in some respect: Note that in that example, the coordination game is "coordination symmetric" in the sense that $x = y$; for such games, Theorem 4.6 can be strengthened to show that the price of stability never exceeds $\frac{3}{2}$; see [MP16] for more details.

Chapter 5

Contagion in Social Networks

Let us now return to the question of how the adoption of a product *spreads* through the network. In particular, we are interested in studying when a product spreads to the *whole* network. We start by analyzing this problem in the simpler model of *plain* networked coordination games (without intrinsic values and without weighted edges), but then remark how the analysis extends also to the more complex models.

5.1 Cascades

Recall that in the plain model, a node decides to choose X if the fraction of its neighbors choosing X exceeds some global adoption threshold $t = \frac{y}{x+y}$. We are interested in determining when X will spread to the entire network; as we observed in Figure 4.1, there exist networks with equilibria where both X and Y are played (and thus, even if players best respond, X will never take over the whole network).

The Contagion Question. If we "infect" an initial set of nodes—think of these as "early adopters" of the product—with a choice of X so that they will choose X *no matter what*, will X spread to the entire network if players follow best-response dynamics? (For instance, what if we decide to promote our Android phone by giving a small number of "influential" people one for free?) We refer to such a spread as a *cascade*; let us formalize this notion.

Definition 5.1. Given a plain networked coordination game $\mathcal{G}$ induced by a graph $G = (V, E)$ and coordination utility Q, we say that a set $S \subseteq V$ is **cascading** with respect to $\mathcal{G}$ if the following process *always* ends with *all* nodes in V choosing X:

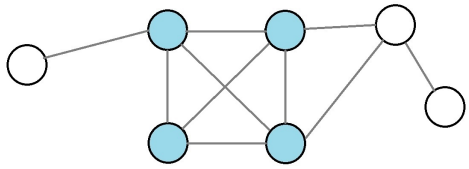


Figure 5.1: The blue nodes form a $\frac{3}{4}$-dense set: every node in this set has at least $\frac{3}{4}$ of its neighbors in the set.

- Start off with an outcome $\vec{a} = (X_S, Y_{-S})$, where every node in S chooses X, and all other nodes chooses Y.
- Run BRD from $\vec{a}$ but where the process is restricted to only nodes in $V \setminus S$ best responding (i.e., nodes in S never change from X, but the others may change strategies by best responding).

Note that since BRD *always* converges in $\mathcal{G}$, it must also converge if we restrict the dynamic to only allowing a subset of the players to best respond—every best-response sequence with respect to the restricted set of players is clearly also one with respect to the full set of players (so if there exists some "loop" w.r.t. the restricted set, such a loop also exists w.r.t to the full set). Thus, the above contagion process (where the BRD is restricted to only the players in $V \setminus S$) will always terminate.

5.2 Characterizing Cascades

We now present a simple condition that exactly characterizes when a set S is cascading. To do this, we will define a notion of the *density* of a set of nodes S.

Definition 5.2. Given an undirected graph $G = (V, E)$ and a set of nodes $S \subseteq V$, we say that S has **density** t if for every node $v \in S$, the fraction of v's neighbors that are inside S is at least t; that is, for all $v \in S$,

$$\frac{|N(v) \cap S|}{|N(v)|} \geq t$$

Theorem 5.3. *Given a plain networked coordination game $\mathcal{G}$ induced by $G = (V, E), Q$ with adoption threshold t, a set S is cascading w.r.t. $\mathcal{G}$ if and only if there does not exist a set of nodes $T \subseteq V \setminus S$ having density $1 - t$ (w.r.t. G).*

Proof. We prove each direction separately.

The "only-if" direction: Assume for contradiction that S is cascading yet the network contains a set $T \subseteq V \setminus S$ of density $1 - t$. Consider the first round in the BRD process when some node $v \in T$ becomes "infected" (i.e., switching to X). At this point, all other nodes in T play Y (since T does not have any intersection with S); thus, by the density requirement of T, the fraction of v's neighbors that are playing Y is at least $1 - t$. Consequently, at most a t fraction of v's neighbors play X, which contradicts that v would switch.

The "if" direction: Consider some set S that is not cascading. We show that the network must contain a set $T \subseteq V \setminus S$ of density $1 - t$. As noted above, the cascade process (i.e., BRD restricted to players in $V \setminus S$) always converges in the game. Consider some *final* outcome of the process where not everyone switched to X (such an outcome must exist since S is not cascading) and let T be the set of nodes playing Y in this outcome. Since nodes in S never switch actions (and always play X), $T \subseteq V \setminus S$. Let us now argue that T must have density $1 - t$. In fact, if it did not, some node $v \in T$ has a greater than t fraction of its neighbors outside of T and would thus want to switch (since by construction all nodes outside T play X), so the outcome could not be final. ■

An interesting consequence of the above theorem is that the *order* of the players in the cascade process (i.e., the restricted BRD process) is irrelevant in determining whether or not a set S cascades!

The Computational Complexity of Finding the Small Cascading Sets Ideally, we would like to have a computationally efficient way of finding a small cascading set. It turns out that this problem is computationally intractable (technically, NP-hard), as shown by [KKT03]. However, in practice, the "greedy" strategy of sequentially infecting players that increase the cascade as much as possible appears to work well (although it may fail miserably on worst-case instances).

5.3 Strong Cascades

The notion of a cascading set assumes that the original set S of "early adopters" never changes actions—i.e. they do not participate in the BRD. We can consider an even stronger notion of cascading where the early adopters only need to start off playing X, but then may themselves participate in BRD (including potentially switching to the choice Y if many of their neighbors are playing it).

Definition 5.4. Given a plain networked coordination game $\mathcal{G}$ induced by a graph $G = (V, E)$ and coordination utility Q, we say that a set $S \subseteq V$ is **strongly cascading** with respect to $\mathcal{G}$ if BRD from the outcome (X_S, Y_{-S}) *always* ends with *all* nodes in V choosing X.

The following theorem provides a sufficient condition for a set of nodes to be strongly cascading.

Theorem 5.5. *Given a plain networked coordination game $\mathcal{G}$ induced by $G = (V, E), Q$ with adoption threshold t, a set S is strongly cascading w.r.t. $\mathcal{G}$ if*

1. *S has density t, and*
2. *there does not exists a set of nodes $T \subseteq V \setminus S$ having density $1-t$ (w.r.t. G).*

Proof. Consider some set S of density t. By the same argument as in the proof of Theorem 5.3, nodes in S will never change from playing X (as at least a fraction t of their neighbors are in S and hence playing X at all times). Hence, running BRD from (X_S, Y_{-S}) is equivalent to running BRD from (X_S, Y_{-S}) but restricting the best-responding players to $V \setminus S$, and so in this particular case S is strongly cascading if and only if it is cascading, which by Theorem 5.3 concludes the proof. ■

An important interpretation of this theorem is that if you want to introduce a new product with the goal of it cascading, carefully pick the initial set of nodes S to which to promote the product so that a) S forms a sufficiently dense cluster (or else, they may decide to switch back to the old product) and b) there is no sufficiently dense cluster of users outside of S.

5.4 Dealing with Subjective Thresholds

So far we have only considered *plain* networked coordination games. Let us turn to analyzing also more general ones. Recall that for the case of plain

networked coordination game, *each* player decides to switch to X if the fraction of its neighbors choosing X exceeds some *global* adoption threshold t. As mentioned, for more general networked coordination games, this no longer holds; rather each node v has their own *subjective* adoption threshold $t(v)$. The results for cascading and strongly cascading sets are easily extended to this setting by considering a more general notion of density, where a set S is said to have **density** $t(\cdot)$ if, for each node $v \in S$,

$$\frac{N(v) \cap S}{N(v)} \geq t(v)$$

In fact, we may further generalize this notion to also deal with networked coordination games with weighted edges by considering a notion of weighted density: a set S is said to have **weighted density** $t(\cdot)$ if, for each node $v \in S$,

$$\frac{\sum_{j \in N(i) \cap S} w(i,j)}{\sum_{j \in N(i)} w(i,j)} \geq t(v)$$

All the results on contagion still apply to networked coordination games with weighted edges, if we replace density with weighted density in the theorem statements.

Notes

Contagion in social networks was first studied by Morris [Mor00]; Theorem 5.3 is a slight variant of a Theorem from [Mor00]. The treatment of strong cascades is new.

Part II

Markets on Networks

Chapter 6

More on Graphs: Flows and Matchings

In this chapter, we consider some additional concepts and problems in graph theory, and discuss some classical results and algorithms. These concepts and algorithms will be useful in the sequel.

6.1 The Max-Flow Problem

Consider a directed graph (V, E) where we assign an integer weight $c(e)$ to each edge of the graph—we will refer to this weight as the *capacity* of the edge e. We refer to the triple $G = (V, E, c)$, where (V, E) is a graph and $c : E \to \mathbb{N}$, as a **weighted graph**.

Given a weighted graph G and two nodes s, t—where s is a *source* (i.e., there are no ingoing edges to s) and t is *sink* (i.e., there are no outgoing edges from t)—consider the problem of finding the *maximum flow* from s to t in G—that is, the maximum amount of some imaginary resource (water, electricity, traffic etc.) that we can route from s to t respecting the capacity of each edge. More formally,

Definition 6.1. A (s, t)**-flow** in a weighted graph $G = (V, E, c)$ where s is a source and t is a sink, is a function $f : E \to \mathbb{R}$ such that the following conditions hold:

- *Conservation of flow:* For every node $v \neq s, t$,

$$\sum_{u|(u,v)\in E} f(u, v) = \sum_{w|(v,w)\in E} f(v, w)$$

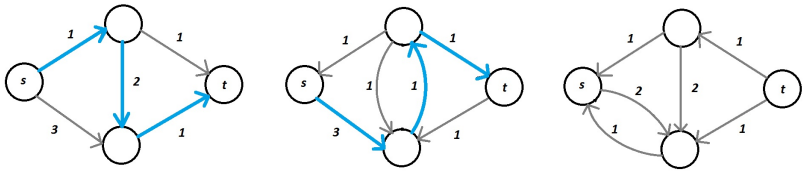


Figure 6.1: Augmenting paths in action: At each stage, we route 1 unit of flow along the augmenting path indicated in blue. Observe that at the end there exist no more augmenting paths, so we have found the maximum flow.

That is, the sum of the flows over incoming edges is the same as the flow over outgoing edges;

- *Respecting Capacity contraints:* For every edge $e \in E$, $f(e) \leq c(e)$.

The **value** of an (s,t)-flow f is the sum of flow coming out of s—that is, $\sum_{v|(s,v)\in E} f(s,v)$. An (s,t)-flow f is said to be a **max-(s,t)-flow** in G if its value is at least as big as the value of any other (s,t)-flow in G.

We can find the maximum flow in a weighted graph G using the **Augmenting Path Algorithm**:

- Find a path P from s to t in the directed graph (assume all edges of capacity 0 are removed).
- "Push" as much flow as possible along this path (i.e. for each edge $e \in P$, add flow equal to $\min_{e\in P} c(e)$).
- Create a new "residual graph" G', constructed as follows:
  - For each edge $e \in P$ where flow was added, decrease its capacity by the same amount as the flow that was added.
  - In addition, for each edge $e = (u,v) \in P$ where flow was added, add the same amount of capacity that was removed from (u,v) to a *reverse* edge (v,u). (This represents that flow can now be routed in the reverse direction—i.e. removed from the current flow; think of this as "backtracking".)
- Reiterate this process on the new graph G', and continue until no further *augmenting path* P can be found.
- The final maximum flow is obtained by summing up the flows picked in each iteration.

See Figure 6.1 for an illustration of the algorithm. It is instructive to note

(formally by induction over the number of iterations) that the final flow output (i.e., the sum of the flows) actually is a flow in the original graph; thus the algorithm produces a correct output.

It is also instructive to note the importance of adding reverse edges: while we would still get a valid flow even if we had not added those edges, they are needed to make sure the algorithm finds the maximum flow. If not, as the example in Figure 6.1 shows, the algorithm gets stuck after the first step. The reverse edges are needed to "backtrack" (i.e., send back) some flow, to finally reach the maximum flow. Let us now argue that the algorithm indeed does find the maximum flow.

6.2 The Max-Flow Min-Cut Duality

To this end, we need to introduce some new definitions that will allow us to assert the correctness of this procedure.

Definition 6.2. An (s,t)**-cut** of a weighted graph $G = (V, E, c)$ is a partition of V into two sets (S, T) such that $s \in S$ and $t \in T$. The **capacity** of an (s,t)-cut (S,T) is the sum of the capacities of all the edges $e = (u, v) \in E$ such that $u \in S$ and $v \in T$ (i.e. all edges leaving S). An (s,t)-cut (S,T) is said to be a **min-**(s,t)**-cut** in G if its capacity is smaller, or equal to, the capacity of any other (s,t)-cut in G.

Theorem 6.3. *Given a weighted graph $G = (V, E, c)$ and a source-sink pair s, t, the Augmenting Path Algorithm outputs some max-(s,t)-flow in G. Furthermore,*

1. *The output flow is always integral;*
2. *The algorithm's running time is polynomial in the size of G and the value of the max-(s,t)-flow.*
3. *The value of the max-(s,t)-flow in G is equal to the capacity of any min-(s,t)-cut in G.*

Proof. First, let us start by observing that since we are assuming capacities are integers, in each step we must be increasing the overall flow in the graph by at least 1, and thus keeping it integral, whenever we find an augmenting path. It follows that the number of iterations is bounded by the capacity of the maximum flow, which thus means that the overall running time is polynomial in that capacity—and that the final output flow is integral. Finally, if we can prove that the output flow is a maximum flow, we have proven the main claim

of the theorem, as well as items 1 and 2. We now prove this and along the way also prove item 3.

Observe that the value of the max flow must be less than or equal to the capacity of the min-cut in G, since any flow traveling from s to t must pass through the edges leaving S. Hence, it suffices to show that the flow from s to t after the procedure is equal to the capacity of *any* (s,t)-cut, since we have argued that the capacity of the min-cut is the highest possible.

Recall that the procedure ends when there are no more augmenting path from s to t. Consider the final residual graph G' at this point. Let S denote the set of nodes that are reachable from s (through edges with positive capacities). Let T be the remaining nodes.

- Clearly, $s \in S$; because t is unreachable from s (or else the algorithm would not have ended), $t \in T$. This (S,T) is an (s,t)-cut.
- All *outgoing* edges from S must have remaining capacity 0 (or else more nodes can be reached from S, and thus S is not the set of nodes that are reachable from s). Hence, the amount of flow we have routed through those edges must be equal to the capacity of those edges in the *original* graph (or else there would be capacity left in the residual graph).
- Finally, observe that there cannot be any flow on *incoming edges* to S, since any such flow can always be "reversed" and thus there would be an outgoing edge with positive capacity (which we have already argued there is not).

Thus we have shown that the flow output by the algorithm equals the capacity of some (s,t)-cut (S,T), and thus it must be equal to capacity of any (s,t)-min cut. We conclude that the algorithm found the max-flow (since as noted above, the value of the max-flow $\leq$ capacity of the min-cut), and in so doing have proven property 3. ■

We mention that special instantiations of the augmenting path algorithm (where we are more careful about which augmenting path P the algorithm selects in each iteration) have an even faster running time for large capacities [EK72], but this will not be of importance to us in this course.

6.3 Edge-Disjoint Paths

Let us now observe a simple application of the max-flow algorithm to the problem of finding *disjoint* paths between nodes s and t in a graph: We say

that two paths from s to t are *edge-disjoint* if they do not have any edges in common. (This application will not be relevant for the subsequent material, but is interesting in its own right.)

Theorem 6.4. *In any graph G, the number of edge-disjoint paths between a source s and a sink t equals the max-(s,t)-flow in (G,c) where $c(e) = 1$ for all edges e. Additionally, there exists a algorithm that finds them whose running time is polynomial in the number of such paths.*

Proof. Note that the number of disjoint paths must be at least as large as the max-flow (since we can just push a flow of 1 on each such disjoint path). Let us show that if the max (s,t)-flow is k then we can also find k disjoint paths; this concludes the proof of the theorem. Assume G has a max (s,t)-flow of k; by Theorem 6.3 we can thus efficiently find an *integral* (s,t)-flow $f : E \to N$ with value k. Since the capacities are all 1, the flow on each edge must thus be either 0 or 1. We now extract out k-disjoint paths from it as follows:

1. Consider the residual graph G' induced by the flow f; that is, remove all edges in G that do not have any flow on them.

2. Find the shortest path from s to t in this graph. If the value of the (s,t) flow is positive, such a path must exist (or else the min-cut would be 0, thus contradicting the max-flow min-cut correspondence of Theorem 6.3). Additionally, note that in the *shortest* path, we never traverse the same vertex (and thus also potentially the same edge) twice (if we did, we would have a loop that could be removed to make the path shorter). Output the shortest path P.

3. Create a new flow f' by decreasing the flow f on all edges along P by 1; note that this is a valid flow (it respects the conservation of flow condition) and the value of it is $k' = k - 1$. As long as $k' \geq 1$, go back to step 1 with the updated flow f'.

The above procedure thus outputs k simple paths (which do not contain any loops). Additionally, since at each step we remove the flow along the output path, we are removing the edges along this path in the residual graph G', and none of these k paths can thus have any edges in common. ■

6.4 Bipartite Graphs and Maximum Matchings

Let us now turn to a seemingly different problem: that of finding *matchings* in *bipartite graphs*. A bipartite graph is simply a graph where the nodes are

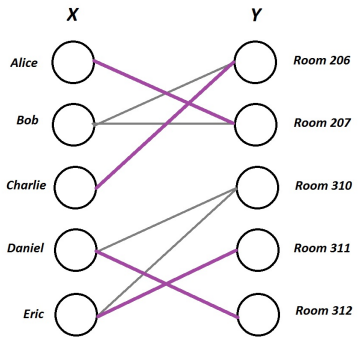


Figure 6.2: The edges in purple constitue a maximum matching (of size 4) between professors and classrooms in the bipartite graph.

divided into two types—"left nodes" X and "right nodes" Y and edges can only exist between a node of one type and a node of the other type. See Figure 6.2 for an example; here the left nodes corresponds to a set of professors, the right nodes corresponds to a set of classrooms, and we draw an edge between a professor and the classroom if the classroom is acceptable to the professor.

Formally,

Definition 6.5. A **bipartite graph** is a graph $G = (V, E)$ where $V = X \cup Y$ and $(x, y) \in E$ only if $x \in X$ and $y \in Y$.

Let us turn to defining the notion of a matching; we provide a general definition that applies to all graphs (and not just bipartite ones), but we shall here focus mostly on bipartite graphs.

Definition 6.6. A **matching** in an undirected graph (V, E) is a set of edges $M \subset E$ such that every node $v \in V$ has at most degree 1 with respect to M (i.e. at most 1 edge in M connected to it). We define the **size** of the matching as $|M|$. We say M is a **maximum matching** if its size is at least as large as the size of any other matching M' in G.

An example of a matching in a bipartite graph can be found in Figure 6.2.

We now show how to use the above Max-Flow algorithm to efficiently find maximum matchings in bipartite graphs.

Theorem 6.7. *There exists a polynomial-time algorithm to find the maximum matching in any undirected bipartite graph G.*

Proof. Given an undirected bipartite graph $G = (X \cup Y, E)$, create an expanded *weighted directed* graph G' with two extra nodes s and t, edges with capacity 1 from s to every node in X, and edges with capacity 1 from every node in Y to t. (See Figure 6.3 for an illustration.) Set all existing edges from X to Y to have infinite capacity[1]. Then find the maximum flow in this graph G' using the augmenting path algorithm; output the edges between X and Y which have flow on them.

Let us now argue that 1) every flow with value k in G' induces a matching of size k in G, and 2) if there exists a matching of size k in G, then there exists some flow of size k in G' (and thus the algorithm will find it).

1. Since the capacities of the edges connecting s and t to the bipartite graph are 1 (and since there are no edges connecting two nodes in X or two nodes in Y), at most 1 unit of flow can pass through each node in X and Y. By Theorem 6.3, the output max flow is integral; hence, the algorithm outputs a valid matching. Additionally, it directly follows that the size of the matching is the value of the max-flow in G'.

2. If G has a matching of size k, we can easily get a flow of k in G'—simply push a flow of 1 from s to every node in X that gets matched (and through its match in Y, and finally to t).

■

6.5 Perfect Matchings and Constricted Sets

Consider now a bipartite graph $G = (X \cup Y, E)$ where $|X| = |Y| = n$. We are interested in understanding when such a bipartite graph has a *perfect* matching, where every node gets matched.

Definition 6.8. A matching M in a bipartite graph $G = (X \cup Y, E)$ where $|X| = |Y| = n$ is said to be **perfect** if *all* nodes in X get matched in M—that is, M is of size n.

[1]Pedantically, in our formal definition of max-flow problem, capacities need to be integers. In this construction, it suffices to set them to e.g., $|V|$; in fact, for the current proof setting them even to 1 will suffice, but for our later usage of this construction it will be useful that the capacities are sufficiently large.

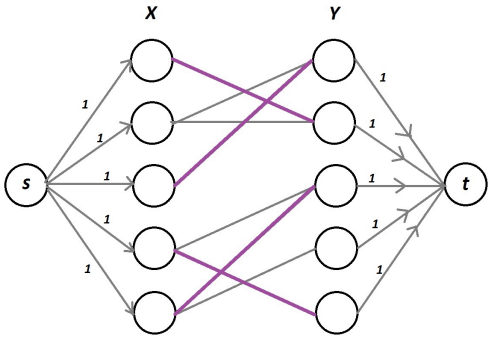


Figure 6.3: The graph from above, expanded to create an s-t flow network. Observe that the maximum flow in this network is equal to the size of the maximum matching.

For instance, if X is a set of n men and Y a set of n women, and the edges (x, y) exist between a man-woman pair if they would find a marriage acceptable; a perfect matching exists if and only if everyone can get "acceptably" married.

Note that we can use the above-described maximum matching algorithm to determine when a perfect matching exists. But if the algorithm notifies us that the maximum matching has size $< n$, it gives us little insight into *why* a perfect matching is not possible.

Constricted Sets The notion of a *constricted set* turns out to be crucial for characterizing the existence of perfect matchings. Given a graph $G = (V, E)$, let $N(S)$ denote the set of neighboring nodes to *any* node in S (that is $x \in N(S)$ if and only if there exists some $y \in S$ and an edge $(y, x) \in E$.) A *constricted set* is a set C of nodes that has less neighbors than elements in C:

Definition 6.9. Given a bipartite graph $G = (X \cup Y, E)$, a **constricted set** is a subset $C \subseteq X$ such that $|N(C)| < |C|$.

Clearly, if a constricted set exists, no perfect matching can be found—a set of k men cannot have acceptable marriages with $< k$ women. The "marriage theorem" proves also the other direction; a perfect matching exists *if and only if* no constricted set exists. We will show not only this theorem, but also how to efficiently find the constricted set when it exists.

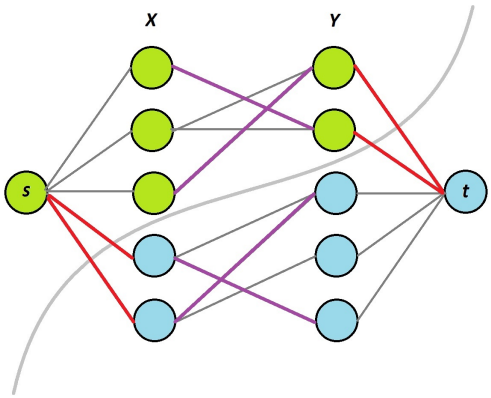


Figure 6.4: An illustration of the construction of a constricted set. Edges in the maximum (not perfect) matching are purple; nodes in S are green, nodes in T are blue, and edges leaving S in the min-cut are red. Observe that $C = X \cap S$, the set of green nodes on the left side, forms a constricted set.

Theorem 6.10. *Given a bipartite graph $G = (X \cup Y, E)$ with $|X| = |Y| = n$, a perfect matching exists if and only if no constricted set exists in G. Additionally, whenever such a constricted set exists, it can be found in polynomial time.*

Proof. As already noted above, if there exists a constricted set, then clearly no perfect matching exists. For the other direction, consider a bipartite graph $G = (X \cup Y, E)$ without a perfect matching. We shall show how to efficiently find a constricted set in this graph, which will conclude the proof of the theorem.

Consider the expanded graph G' from the proof of Theorem 6.7; use the max-flow algorithm from Theorem 6.3 to find the max-(s,t)-flow, and consider the min-(s,t)-cut, (S,T), induced by this flow in the proof of Theorem 6.3. Recall that the set S was obtained by considering the set of nodes that are reachable from s in the final residual graph; this set of nodes can be found in polynomial time using breadth-first search (see Claim 2.12).

We now claim that $C = S \cap X$ is a constricted set.

- Since G does not have a perfect matching, by Theorem 6.7 (i.e., the max-matching/max-flow correspondence) and Theorem 6.3 (i.e., the max-flow/min-cut correspondence) the capacity of the min-cut is strictly less than n. Since the outgoing edges from X to Y have infinite capacity,

none of those edges can connect S to T (otherwise the cut would have infinite capacity).[2]

- Thus, the capacity of the min-cut is the sum of the number of edges from s to $X \cap T$ and the number of edges from $Y \cap S$ to t; see Figure 6.4 for an illustration. In other words, the capacity of the min-cut is

$$|X \cap T| + |Y \cap S| < n$$

where the inequality follows since, as argued above, the min-cut is strictly smaller than n.

- On the other hand, since (S, T) is a partition of $X \cup Y$, it is also a partition of X. Thus,

$$|X \cap T| + |X \cap S| = n$$

- Combining the above two equations, we get

$$|X \cap S| > |Y \cap S|$$

- Finally, note that since (as argued above) the (S, T)-cut does not cut any edges between X and Y (recall they have infinite capacity), we have that $N(X \cap S) \subseteq Y \cap S$. We conclude that

$$|X \cap S| > |Y \cap S| \geq |N(X \cap S)|$$

which proves that $C = X \cap S$ is a constricted set.

■

Notes

We refer the reader to [CLRS09] and [KT05] for a more in-depth study of graph algorithms. The Augmenting Path Algorithm path algorithm is due to Ford and Fulkerson [FF62]. The "marriage theorem" is due to Hall [Hal35].

[2] In fact, it suffices to set the capacity of the edges from X to Y to n.

Chapter 7

Traffic Network Games

In this chapter, we return to traffic flows, but this time consider them in a game-theoretic context.

7.1 Definition of a Traffic Network Game

A **traffic network game** $\mathcal{G}$ on a directed graph $G = (V, E)$ is specified as follows:

- We associate with each edge $e \in E$, a *travel time function* $T_e(\cdot)$, which determines the travel time $T_e(x)$ on the "road" e if x players are traveling on it.
- We assume the travel time is linear: $T_e(x) = \alpha_e x + \beta_e$, where $\alpha_e, \beta_e \geq 0$. (So, the more people are traveling on an edge, the longer it should take.)
- We specify a source $s \in V$ and a target $t \in V$; the goal of *all* players is to travel from s to t. The action set for each player is the set of paths p from s to t.
- An outcome (action profile) $\vec{p}$ thus specifies a path for each player.
- In such an outcome $\vec{p}$, let $x_e(\vec{p})$ denote the number of players traveling on edge e in that outcome.
- Each player i's utility is defined as the *negative* of their travel time; that is,

$$u_i(\vec{p}) = -\sum_{e \in p_i} T_e(x_e(\vec{p}))$$

  (The reason we take the negative of the travel time is so that longer travel times are considered less desirable.)

More formally,

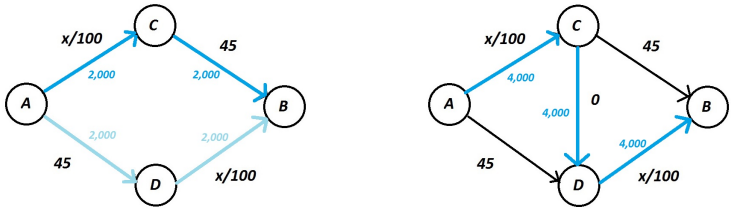


Figure 7.1: Left: A basic example of a traffic network game. Right: What happens to the PNE for this game when we add a road?

Definition 7.1. We refer to the tuple $(G = (V, E), \{\alpha_e, \beta_e\}_{e \in E}, s, t)$ such that $s, t \in V$ as a **traffic network problem**. The **traffic network game** $\mathcal{G} = (n, A, u)$ corresponding to the traffic flow problem $(G = (V, E), \{\alpha_e, \beta_e\}_{e \in E}, s, t)$ is defined as follows:

- For each player i, let A_i be the set of (s, t)-paths in G.
- For each player i, let

$$u_i(\vec{p}) = -\sum_{e \in p_i} T_e(x_e(\vec{p}))$$

where T_e, x_e are defined as above.

Example. Consider the traffic network in the left of Figure 7.1 with 4,000 players traveling from A to B, where there are four edges; $A \to C$ and $D \to B$ have travel time $T_{A \to C}(x) = T_{D \to B}(x) = \frac{x}{100}$, and $C \to B$ and $A \to D$ have travel time $T_{C \to B}(x) = T_{A \to D}(x) = 45$. There are only two possible paths from A to B, and thus only two possible actions for the players: let us call them UP ($A \to C \to B$) and DOWN ($A \to D \to B$). If everyone goes UP, then the travel time for everyone is $40 + 45 = 85$; the same applies if everyone goes DOWN.

But if half go UP and half go DOWN, everyone gets a travel time of $20 + 45 = 65$. This is, in fact, the unique PNE for this game—if more than half of the players are traveling along one of the two paths, a player traveling on that path can decrease their travel time by switching to the other (less traveled) path.

7.2 Braess's Paradox

Consider again the network on the left in Figure 7.1 but let us augment it by adding an extremely efficient road from C to D with travel time 0, resulting in the network on the right in the figure. Intuitively, we would think that adding such an amazing new road would decrease the travel time for everyone. Surprisingly, the opposite happens!

Note that we have added a new path (i.e., action), $A \to C \to D \to B$ to the game; let us call this action HIGHWAY. There is again a unique PNE in the game, and it is one where everyone plays HIGHWAY, which leads to the travel time *increasing* to $40 + 40 = 80$! In fact, HIGHWAY is a strictly dominant action (and thus the only PNE)—$A \to C$ is a strictly better choice than $A \to D$ (even if everyone travels on $A \to C$), and $D \to B$ is a strictly better choice than $C \to B$ (even if everyone travels on $D \to B$).

So, by adding an extra road, we have increased everyone's travel time from 65 to 80! The point is that although adding a new road can never make things worse in the socially optimal outcome (i.e., the outcome maximizing social welfare), the socially optimal outcome may not be an equilibrium; additionally, "earlier" equilibria (of the game without the new road), may also be disrupted since players now have more actions (and thus more ways to deviate to improve their own utility). This is similar to the Prisoner's Dilemma; if we had restricted the game to a single "cooperate" action for both players, we would get a good outcome, but once we add the possibility of "defecting", the only PNE leads to a socially bad outcome.

Let us next consider the traffic network games more generally and answer the following questions:

- Do equilibria always exist in traffic network games?
- Will BRD always converge in them?
- What is the price of stability? (That is, how bad can the "gap", between the socially optimal outcome and the best equilibrium, be?)

7.3 Convergence of BRD

As in Chapter 4, we will show that BRD converge, and hence that PNE exist, by exhibiting a potential function. First of all, let us look at social welfare. Let $T(\vec{p})$ denote the *total travel time* of all players:

$$T(\vec{p}) = \sum_{i=1}^{n} \sum_{e \in p_i} T_e(x_e(\vec{p})) = \sum_{e \in E} \sum_{i \in [x_e(\vec{p})]} T_e(x_e(\vec{p})) \tag{7.1}$$

By definition, the social welfare is the negative of the total travel time; that is

$$SW(\vec{p}) = -T(\vec{p})$$

As may be inferred from the example above, this is not an ordinal potential function (just as was the case with networked coordination games)—consider the game with the new highway, and the outcome where half the players play UP and half DOWN; every player wants to deviate to HIGHWAY which decreases SW. Instead, similarly to networked coordination games, we can define a different potential function based on a variant of SW. Consider, $\Phi(\vec{p}) = -L(\vec{p})$ where L is some variant "travel time energy" defined as follows:

$$L(\vec{p}) = \sum_{e \in E} \sum_{i \in [x_e(\vec{p})]} T_e(i)$$

So, on each edge, instead of counting everyone's *actual* travel time (as in the definition of total travel time), we count the travel time as if the players were to arrive sequentially to the road: the first gets a travel time of $T_e(1)$, the second $T_e(2)$, and so on; we refer to this alternative way of counting the travel time on each edge as the "sequential-arrival travel time".

Claim 7.2. *For any Traffic Network Game G, the potential function Φ defined above is ordinal.*

Proof. Consider an action profile $\vec{p}$ and a player i who can strictly improve their utility by switching to an alternative path p_i'. We wish to show that the "travel energy" L decreases due to this switch (hence, Φ increases)—that is, that

$$L(p_i', p_{-i}) - L(p_i, p_{-i}) < 0$$

Recall that $L(\cdot)$ is defined as the sum of the "sequential-arrivial travel times" along all edges in the graph. Additionally, note that when player i switches from p_i to p_i', the sequential-arrivial travel time is only affected on edges along p_i and p_i'; in fact, it is only affected on edges that are on one path but not the other (i.e., on $p_i \setminus p_i'$ and $p_i' \setminus p_i$).

- Over edges in $p_i \setminus p_i'$, we remove *in total* $\sum_{e \in p_i \setminus p_i'} T_e(x_e(\vec{p}))$ in sequential-arrival travel time (since there is one less individual on those edges).
- Over edges in $p_i' \setminus p_i$, we add *in total* $\sum_{e \in p_i' \setminus p_i} T_e(x_e(\vec{p}) + 1)$ in sequential-arrival travel time (since we add one extra individual on those edges).

But difference is exactly the same as the change in i's *actual* (as opposed to sequential-arrival) travel time, which we know is negative because i's utility strictly increases when changing paths. ■

So, it immediately follows (from the above and from Theorem 3.3) that:

Theorem 7.3. *BRD converge in all Traffic Network Games.*

7.4 Price of Stability

We now turn to analyzing how selfish behavior degrades the efficiency of outcomes; that is, we want to study the price of stability. For games where utilities are non-positive (i.e., games were the utility represents a cost rather than a reward), the price of stability is defined as the ratio between the utility of the best PNE's and the maximum social welfare (as opposed to the ratio between the MSW and the utility of the best PNE as we defined it in games with non-negative utilities).

Definition 7.4. The **Price of Stability** in a game $\mathcal{G}$ with non-positive utilities is defined as

$$\frac{\max_{\vec{a} \in PNE^{\mathcal{G}}} SW^{\mathcal{G}}(\vec{a})}{MSW^{\mathcal{G}}}$$

where $PNE^{\mathcal{G}}$ denotes the set of PNE in $\mathcal{G}$.

We show that the price of stability is at most 2 in traffic network games; that is, the travel time in equilibrium cannot be more than a factor 2 worse than the socially optimal travel time.

Theorem 7.5. *In every Traffic Network Game G, there exists a PNE $\vec{p}'$ such that*

$$\frac{SW^{\mathcal{G}}(\vec{p}')}{MSW^{\mathcal{G}}} \leq 2$$

(In other words, the Price of Stability is at most 2.)

Proof. Showing the theorem amounts to proving that there exists a PNE $\vec{p}'$ such that

$$T(\vec{p}') \leq 2 \min_{\vec{p}} T(\vec{p})$$

We proceed similarly to the proof of our bound for the price of stability in coordination games (i.e. Theorem 4.6). We first show that L "approximates" T, and then use this to deduce the bound.

Claim 7.6. *For every action profile $\vec{p}$,*

$$T(\vec{p}) \geq L(\vec{p}) \geq \frac{1}{2}T(\vec{p})$$

Proof. Clearly, by Equation 7.1, we have that $T(\vec{p}) \geq L(\vec{p})$. Additionally, we claim that $L(\vec{p}) \geq \frac{1}{2}T(\vec{p})$. Below, whenever $\vec{p}$ is clear from context, we abuse of notation and simply let x_e denote $x_e(\vec{p})$. Recall that,

$$L(\vec{p}) = \sum_{e \in E} \sum_{i \in x_e(\vec{p})} T_e(i) = \sum_{e \in E} \left(\sum_{i \in x_e} (\alpha_e i + \beta_e) \right) = \sum_{e \in E} \left(\alpha_e \sum_{i \in x_e} i + x_e \beta_e \right) =$$

$$= \sum_{e \in E} \left(\frac{x_e(x_e+1)}{2} \alpha_e + x_e \beta_e \right) \geq \frac{1}{2} \sum_{e \in E} \left(x_e^2 \alpha_e + x_e \beta_e \right) = \frac{1}{2} \sum_{e \in E} x_e T_e(x_e) = \frac{1}{2} T(\vec{p})$$

as desired. ■

So, just as in our coordination game proof (Theorem 4.6), pick some state $\vec{p}$ that minimizes $T(\vec{p})$, and run BRD until we arrive at a final state $\vec{p}'$; by Theorem 7.3 such a state must exist, and it is a PNE. Since L decreases at every step in this process (by Claim 7.2), we have

$$T(\vec{p}) \geq L(\vec{p}) \geq L(\vec{p}') \geq \frac{1}{2}T(\vec{p}')$$

which concludes the proof. ■

Notes

Traffic network games as we described them here were first studied by the Rosenthal [Ros73], who also showed the existence of PNE and convergence of BRD; Rosenthal also introduced the potential function that we consider here. Braess paradox was first described in [Bra68]; our description of Braess' paradox follows the treatment in [EK10].

As already discussed in Chapter 4, the "inefficiency" of equilibria was first studied in [KP09]; [KP09] also explicitly considered the inefficiency of equilibria in traffic network games, but in a different model than we consider here. Roughgarden and Tardos [RT02] were the first to study the inefficiency of equilibria in network traffic games (as we consider them here) and proved a tight bound of $\frac{4}{3}$ for the price of stability (in fact even for the Price of Stability) in a slightly different "non-atomic" model where it is assumed that users control only a negligible fraction of the overall traffic load. The price of stability bound of 2 for the "atomic" case which we presented here is also due to [RT02] but (as far as we know) was first published in [EK10];

Chapter 8

Matching Markets

Let us now return to the second example in the introduction and to the bipartite matching problems we considered in Chapter 6. Consider the bipartite graph in Figure 6.2 (which is the same one we considered in the introduction) where we have a set of three people (A, B, and C) on the left side and a set of three houses (H_1, H_2, and H_3) on the right side, and interpret the existence of an edge between an individual and a house as saying that the individual finds the house *acceptable*.

Using the maximum matching algorithm described in Chapter 6, we can find the largest set of players that can be "matched" (i.e., assigned) a house that they find acceptable (specifically, one such maximum matching would match A to H_2, B to H_1 and C to H_3). But, in any such maximum matchings, we are only guaranteed that the people get some house they find "acceptable". In reality, however, people have *preferences* over houses; we may represent these preferences by specifying, for each player, a *(subjective) valuation* for each house; we refer to such a situation as a **matching market**.

Additionally, houses are typically not free; we may thus also consider a setting where we associate *prices* with each house; we refer such a setting as

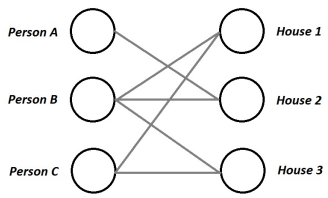


Figure 8.1: A basic example of a matching problem between people and houses.

a **priced matching market**. In this chapter, we will study such (priced) matching markets.

8.1 Definition of a Matching Market

Given a set of players (buyers) $X = [n] = \{1, \ldots, n\}$ and a set of n items Y:

- we associate with each player (buyer) $i \in X$, a *valuation function* $v_i : Y \to \mathbb{N}$. For each $y \in Y$, $v_i(y)$ determines i's value for item y.
- we associate with each item $y \in Y$, a price $p(y) \in \mathbb{N}$.
- a buyer i who receives an item y gets utility

$$v_i(y) - p(y)$$

  (that is, the valuation it has for the item, minus the price it needs to pay); buyers that do not get any item get utility 0.

For simplicity of notation, we restrict to the case when $|X| = |Y|$, but the more general case can be reduced to this special case by either adding "dummy buyers" (who value all items at 0) or adding "dummy items" (with value and price 0).

Definition 8.1. We refer to the tuple $\Gamma = (n, Y, v)$ (as defined above) as a **matching market**, and the tuple $(\Gamma, p) = (n, Y, v, p)$ as a **priced matching market**.

8.2 Acceptability and Preferred Choices

We say that item y is **acceptable** to buyer i if $v_i(y) \geq p(y)$ (i.e. buyer i has value for item y that is at least its market price). We can now define the "acceptability graph" induced by a priced matching market:

Definition 8.2. Given a priced matching market (n, Y, v, p), we define the **induced acceptability graph** $G = ([n] \cup Y, E)$, where $(i, y) \in E$ if and only if y is acceptable to i.

The fact that y is acceptable to i does not imply that i prefers y over all other items; we say an item y is a **preferred choice** for i if y is acceptable to i *and* y maximizes $v_i(y) - p(y)$ over all items in Y—that is,

$$v_i(y) - p(y) = \max_{y' \in Y} v_i(y') - p(y')$$

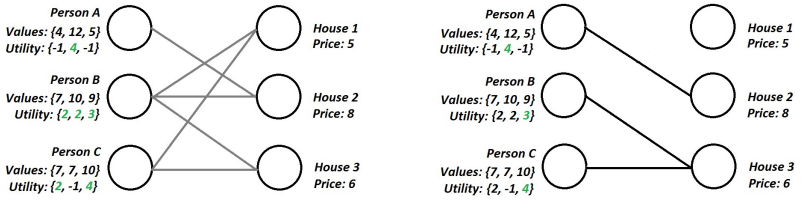


Figure 8.2: Left: An acceptability graph for an example priced matching market with the given prices and values. Buyers are linked to any houses for which they have non-negative utility. Right: The preferred choice graph for the same example. Note that now each buyer is linked only to houses for which they have the *highest* utility.

Note that there may not necessarily exist a unique preferred choice for all buyers; for instance, two items could have the same valuation and price.

The notion of a preferred choice allows us to construct a different bipartite graph:

Definition 8.3. Given a priced matching market (n, Y, v, p), we define the **induced preferred choice graph** as $G = ([n] \cup Y, E)$, where $(i, y) \in E$ if and only if y is a preferred choice to i.

See Figure 8.2 for an illustration of induced acceptability and preferred choice graphs. We can easily guarantee a perfect matching in the acceptability graph by making all items free ($p(y) = 0$ for all $y \in Y$), since then every item would be acceptable to every buyer. (So we can just give the item with index i to buyer i.) Can we also set prices so that everyone ends up with an outcome they are the *most happy* with (and not just find acceptable)? That is, can we set prices so that there exists a perfect matching in the preferred choice graph (and thus everyone ends up with their preferred choice)? The notion of *market clearing* and *market equilibrium* addresses this.

Definition 8.4. Given a matching market $\Gamma = (n, Y, v)$, we say that prices $p : Y \to \mathbb{N}$ are **market-clearing** for Γ if there exists a perfect matching M in the preferred choice graph induced by (Γ, p); we refer to M as the **market-clearing matching**, and the pair (p, M) as a **market equilibrium** (or **Walrasian equilibrium**).

8.3 Social Optimality of Market Clearing

Whereas a matching in the acceptability graph by definition ensures that each buyer i who gets matched to an item y gets non-negative utility, a matching M in the preferred choice graph also ensures that the assignment of buyers to items *maximizes* the *social value* of the items—that is, the items are allocated to buyers in a way that maximizes the total value they generate.

Let us formalize this: An *assignment* $\alpha : X \to Y \cup \perp$ is a function that, given a buyer $i \in X$, outputs either an item $y \in Y$ (meaning i was assigned item y) or $\perp$ (meaning that i was not assigned an item), such that no two buyers $i, j \in X$ get mapped to the same item $y \in Y$ (i.e., $\alpha(i) = \alpha(j) = y$). Note that any matching M induces an assignment α_M, where $\alpha_M(i)$ simply is the item to which i is matched in M (or $\perp$ if i is not matched); to simplify notation, we abuse of notation and let $M(i)$ denote $\alpha_M(i)$.

We can now define the **social value** of an assignment in a matching market Γ as

$$SV^{\Gamma}(\alpha) = \sum_{i \in [n], \alpha(i) \neq \perp} v_i(\alpha(i))$$

Note that this notion of social value only takes into account the value the buyers have for the items; the model implicitly assumes the seller has no value for the items. This is actually without loss of generality: if the seller has some values $v_S(y)$ for items $y \in Y$, we can capture this by introducing an additional "buyer" (representing the seller's interests in keeping the items) with those valuations.

Note the social value is different from sum of buyer's *utilities*, as it considers only the *value* the buyers get from the items, but without considering the prices they need to pay for them. However, it is not hard to see that the social value of an assignment actually equals the social welfare of *all players* (i.e., if we consider the utility of both the buyers and the sellers) *no matter what the prices are*. This follows from the fact that any decrease in utility imposed on the buyers due to the payment, is offset by the increase in utility for the sellers (as they receive those payments):

$$\begin{aligned} SW^{\Gamma}(\alpha, p) &= \text{buyers' utilities} + \text{sellers' utilities} \\ &= \sum_{i \in [n], \alpha(i) \neq \perp} u_i(\alpha(i)) + \sum_{i \in [n], \alpha(i) \neq \perp} p(\alpha(i)) \\ &= \sum_{i \in [n], \alpha(i) \neq \perp} (v_i(\alpha(i)) - p(\alpha(i))) + \sum_{i \in [n], \alpha(i) \neq \perp} p(\alpha(i)) \end{aligned}$$

$$= \sum_{i \in [n], \alpha(i) \neq \perp} v_i(\alpha(i)) = SV^{\Gamma}(\alpha)$$

We thus have the following claim:

Claim 8.5. *For all α, p, $SW^{\Gamma}(\alpha, p) = SV^{\Gamma}(\alpha)$.*

We say that an assignment α **maximizes social value** if $SV^{\Gamma}(\alpha) = \max_{\alpha'} SV^{\Gamma}(\alpha')$. Similarly, an assignment α and and price function p **maximize social welfare** if $SW^{\Gamma}(\alpha, p) = \max_{\alpha', p'} SW^{\Gamma}(\alpha', p')$. By Claim 8.5, we directly have the following corollary.

Corollary 8.6. *For any matching market $\Gamma = (n, Y, v)$, an assignment α maximizes social value if and only if (α, p) maximize social welfare for any (or every) p.*

Thus, if we have an outcome that maximizes social value, then social welfare will be maximized *regardless of how we set prices*.

Let us now use this observation to show that any market equilibrium maximizes social welfare.

Theorem 8.7 (Social optimality of market equilibria.)**.** *Given a matching market Γ and a market equilibrium (p, M) for Γ, we have that α_M (i.e., the assignment corresponding to the matching M) maximizes social value.*

Proof. Consider some market equilibrium (p, M) for Γ; we aim to show that α_M maximizes social value. Towards this, consider some (potentially other) assignment α' that maximizes social value for Γ. By Corollary 8.6, (α', p) also must maximize social welfare (in fact, α', p' maximizes social welfare for any p'). Since M is a perfect matching in the preferred choice graph, by assumption, *every* buyer must receive an item that maximizes his own utility given the prices. Thus,

$$SW^{\Gamma}(\alpha_M, p) \geq SW^{\Gamma}(\alpha', p)$$

since each buyer does at least as well in α_M as in α', and as all items are sold in α_M, the sellers' utilities are maximized *for the fixed set of prices p* as well. Thus, (α_M, p) must also maximize social welfare, and then Corollary 8.6 we have that α_M maximizes social value as desired. ■

Let us end this section by noting that whereas the proof of Theorem 8.7 is relatively simple, the fact that this result holds is by no means a triviality—we have already seen several examples of situations (e.g., coordination and traffic

games) where equilibria lead to outcomes that do not maximize social welfare. Theorem 8.7 is an instance of what is known as *the first fundamental theorem of welfare economics* and a formalization of Adam Smith's famous *invisible hand* phenomenon—that free markets lead to a socially optimal outcome as if guided by "unseen forces".

8.4 Existence of Market Clearing Prices

So, given the awesomeness of market equilibria, do they always exist? In fact, it turns out that they always do! Thus, we can always set prices in such a way that 1) everyone gets one of their preferred choices, and 2) items are given to people in a manner that maximizes social value.

We now show that not only market equilibria always exists, but they can also be efficiently found (relying on the material covered in Chapter 6). This theorem is an instance of what is referred to as *the second fundamental theorem of welfare economics.*

Theorem 8.8. *A market equilibrium (p, M) exists in every matching market Γ. Additionally, there exists an efficient procedure which finds this equilibrium in time polynomial in the maximum valuation V that any buyer has for some item.*

Proof. We present a particular *price updating mechanism*—analogous to BRD for games—which will converge to market clearing prices. Start by setting $p(y) = 0$ for all $y \in Y$. Next, *iteratively* update prices as follows:

- If there exists a perfect matching M in the preferred choice graph, we are done and can output (p, M). Recall that by Theorem 6.7, this step can be performed efficiently.
- Otherwise, by Theorem 6.10, there exists a *constricted* set of buyers $S \subseteq [n]$ (i.e. a set such that $|N(S)| < |S|$) in the preferred choice graph; furthermore, recall that this constricted set may also be found efficiently.
- Next, raise the prices of all items $y \in N(S)$ by 1.
- If the price of *all* items is now greater than zero, *shift* all prices downwards by the same amount (i.e. 1, since we can only increase prices by one at a time) to ensure that the cheapest item has a price of zero.

(An example of the execution of this procedure can be found in Figures 8.3 and 8.4.) Clearly, if this procedure terminates, we have found a market equilibrium. Using a potential function argument, we can argue that the

Prices:	{0, 0, 0}	{0, 1, 0}	{0, 2, 1}	{0, 3, 2}
Person A Values: {4, 12, 5}	{4, 12, 5}	{4, 11, 5}	{4, 10, 4}	{4, 9, 3}
Person B Values: {7, 10, 9}	{7, 10, 9}	{7, 9, 9}	{7, 8, 8}	{7, 7, 7}
Person C Values: {7, 7, 10}	{7, 7, 10}	{7, 6, 10}	{7, 5, 9}	{7, 4, 8}
Constricted Set	{A, B} prefer 2	{A, B, C} prefer {2, 3}	{A, B, C} prefer {2, 3}	Done!

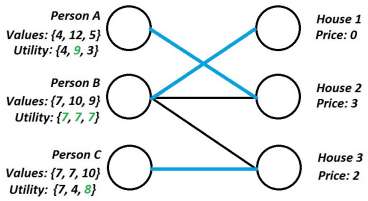


Figure 8.3: The algorithm detailed here when applied to the matching market above. The perfect matching on the final preferred-choice graph (i.e. the equilibrium assignment) is shown in blue.

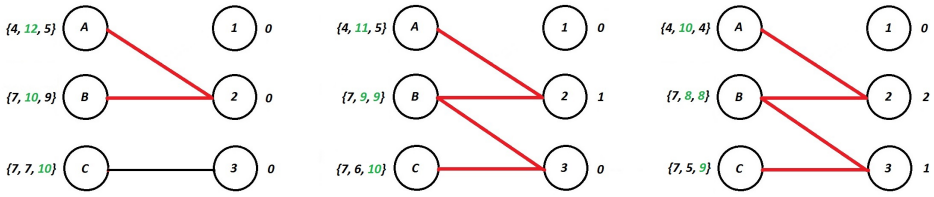


Figure 8.4: The preferred choice graphs for the intermediate states of the algorithm. Constricted sets are indicated by red edges.

procedure does, in fact, always terminate (and thus always arrives at a market equilibrium):

- For each buyer $i \in [n]$, define $\Phi_i^B(p)$ as the *maximal utility buyer i can hope to get*; that is, $\max_y(v_i(y) - p(y))$. (Note that, if we have a perfect matching in the preferred choice graph, this is actually the utility buyer i gets; but if no such perfect matching exists, some buyer must get less.)
- Let $\Phi^B(p) = \sum_{i \in [n]} \Phi_i^B(p)$ be the sum of the potentials for all buyers.
- Let $\Phi^S(p) = \sum_y p(y)$ be the maximal utility the sellers can hope to get. (Again, note that this utility is attained in the case of a perfect matching, but may not be otherwise.)
- Define the final potential as $\Phi(p) = \Phi^B(p) + \Phi^S(p)$.

Let us first show that the potential is always non-negative.

Claim 8.9. *Consider a single iteration of the algorithm where prices get updated from p to p'. Then $\Phi(p') \geq 0$.*

Proof. First, note that prices start off non-negative (in fact, zero) and then can never become negative (in particular, we only ever decrease prices until the cheapest item is free). It thus follows that $\Phi^S(p') \geq 0$.

Next, since there always exists some item with price 0, we know that $\Phi_i^B(p') \geq 0$ for every buyer i, since i may always obtain non-negative "potential utility" by going for this free item. So $\Phi^B(p') \geq 0$, and thus $\Phi(p') \geq 0$. ■

We next show that the potential decreases at every iteration of the algorithm.

Claim 8.10. *Consider a single iteration of the algorithm where prices get updated from p to p'. Then $\Phi(p') < \Phi(p)$.*

Proof. Recall that the algorithm first increases the price of the items in the neighbor set, $N(S)$, of the constricted set, S, by 1. This increases Φ^S by $|N(S)|$. But it decreases the maximal utility for each buyer $i \in S$ by 1, since they now must pay 1 extra for their preferred item—we here rely on the facts that a) since there exists some item with price 0, every player i has at least one preferred item, thus $N(S) \geq 1$, and b) since valuations are integers, if increasing the prices of *all* preferred choices of i by 1, the items will still be preferred choices of i. So Φ^B decreases by $|S|$. However, by the definition of a constricted set, $|S| \geq |N(S)| + 1$, and so Φ must decrease by at least 1 during this phase.

Next, if we shift all prices downwards by one, Φ^S will decrease by n, but Φ^B will increase by n (by the same logic as above), and so the price shift step has no impact Φ. ■

Finally, notice that, at the start, $\Phi^S = 0$ and

$$\Phi^B = \sum_{i \in [n]} \max_{y \in Y} v_i(y) \leq nV$$

where V is the maximal valuation any player has for any item. So, by the above two claims, the procedure will terminate, and it will do so in at most nV iterations. ■

8.5 Emergence of Market Equilibria

Tatonnement The above proof just shows that market clearing prices (and hence market equilibria) always exist. But how do they arise in the "real world"? We can think of the process described above as a possible explanation for how they might arise over time. If demand for a set of items is high (there

are too many people who prefer those items—that is, we have a constricted set in the preferred choice graph), then it seems reasonable to assume that the market will adjust itself to increase the prices of those items.

The price shift operation, however, seems somewhat less natural. But it is easy to see that it suffices to shift prices downwards whenever *all* items are too expensive for some buyer (in other words, when the potential utility of some buyer becomes negative) and thus we cannot sell everything off. In such a case, it may seem more natural for the sellers to decrease market prices (e.g., in a housing market, if too many houses remain for sale, the whole housing market starts going down).

In general, these type of market adjustment mechanisms, and variants thereof, are referred to as *tatonnement* (French for "groping") mechanisms and have been the topic of extensive research. Whether market clearing prices and equilibria actually do arise in practice (or whether, e.g., conditions change faster than the tatonnement process converges to the equilibrium) is a question that has been debated since the Great Depression in the 1930s.

Buyer-optimal and Seller-optimal Market Equilibria Let us remark that market equilibria are not necessarily unique. Some equilibria are better for the seller, whereas others are better for the buyers. To understand why this can happen, consider the (degenerate) special case of a single buyer and a single item for sale; if the buyer values the item at v, then *every* price $p \in [v]$ can be sustained in an equilibrium (p, M) (where M is the trivial matching where the buyer gets matched with the seller). In particular, setting the price $p = 0$ is clearly optimal for the buyer, whereas setting the price $p = v$ is optimal for the seller. A similar reasoning can be applied also in more complicated situations: we can often shift prices by a small—or, as in Figure 8.5, a large—amount while preserving the matching in the market; higher prices will be better for the seller, and lower prices will favor the buyer.

8.6 Bundles of Identical Goods

Finally, let us consider a simplified special case of the matching market model, where each item $y_j \in Y$ is a *bundle* of c_j *identical goods* such that the value of bundle y_j to buyer i is:

$$v_i^{\vec{c},\vec{t}}(y_j) = c_j t_i$$

where t_i is player i's *(subjective) value* for a single good. We can assume without loss of generality that the bundles are ordered in decreasing size—that is, $c_1 \geq c_2 \geq \ldots \geq c_n$.

Figure 8.5: Left: The equilibrium we found with the algorithm is, in fact, buyer-optimal. Right: If we increase the prices of all houses by as much as possible such that our matching in the preferred choice graph stays the same (even though the graph itself doesn't!), we obtain the seller-optimal equilibrium. Notice that the seller now gets all of the utility!

Observe that in any outcome that maximizes social value, we have that the largest bundle y_1 must go to the person who values the good (and hence the bundle of goods) the most. Consequently, the person who values the good the second-most should get y_2, and so on and so forth. Hence, by Theorem 8.7 (which states that market equilibria maximize social value), this property must also hold in any market equilibrium.

We now show that in any market equilibrium, the largest bundles will also have the highest price/good (i.e., the highest "unit-price"). As first sight this may seem counter-intuitive—when shopping for groceries we would expect a larger bundle of goods to have a lower price/good! The difference here is that the supply of bundles is limited and each buyer can only purchase a single bundle.[1]

Theorem 8.11. *For every matching market for bundles of identical goods given by* $\Gamma = (n, Y, v^{\vec{c},\vec{t}})$*, and for every market equilibrium* (p, M) *for* Γ*, it holds that for every* $j = 1, \ldots, n-1$*,*

$$\frac{p(y_j)}{c_j} \geq \frac{p(y_{j+1})}{c_{j+1}}$$

[1]The bounded supply assumption is actually only one part of the story here. For instance, another reason why we tend to see lower prices for higher quantity bundles in practice is that values for bundles usually are not strictly linear in the size of the bundle (as we have modelled them here): it is common to have decreasing marginal values.

Proof. Consider a market equilibrium (p, M). Let $\alpha_j = \frac{p(y_j)}{c_j}$ be the price/good in bundle y_j. Consider two bundles $y_j, y_{j'}$ where $j < j'$, and assume for the sake of contradiction that $\alpha_j < \alpha_{j'}$—that is, the goods in the *larger* bundle y_j are *cheaper*. Now consider the player i who is matched to the smaller bundle $y_{j'}$ of more expensive goods. His utility is currently $c_{j'}(t_i - \alpha_{j'})$, which must be non-negative by the market clearing property of (p, M). If he were to switch to the larger bundle y_j of cheaper goods, he could increase his utility by $c_j(t_i - \alpha_j) - c_{j'}(t_i - \alpha_{j'}) > 0$, since $c_j \geq c_{j'}$ and $\alpha_j < \alpha_{j'}$ by assumption. So $y_{j'}$ cannot be a preferred choice for i, contradicting the fact that (p, M) is a market equilibrium. ■

Notes

Matching markets were first considered by Gale and Shapley [GS62] and the formal model of introduced by Shapley and Shubik [SS71]. Shapley and Shubik [SS71] also proved that existence of market equilibria; the constructive procedure for finding market equilibria is due to Demange, Gale, and Sotomayor [DGS86], and the analysis technique (using potential functions) is due to [EK10]. The notion of a market equilibrium (i.e. a Walrasian equilibrium), and the tatonnement process dates back to the 1870s and the work of Leon Walras [Wal54]. (The treatment of bundles of identical goods is new.)

Chapter 9

Exchange Networks

In this chapter, we will consider a generalized form of a matching markets, referred to as *exchange networks*. Roughly speaking, rather than just considering bipartite graphs where the left nodes are buyers and the right nodes are items/sellers, we will now consider interactions/exchanges over arbitrary networks/graphs. We first describe the problem and then relate it back to matching markets.

9.1 Definition of an Exchange Networks

Assume we have a graph $G = (V, E)$ with n nodes; the nodes in this graph represent the players, and an edge between two players means that the players may perform "an exchange" with each other. Each edge e is associated with some amount/value $v(e)$ that can be split between its endpoints if an exchange between the endpoints is taking place—think of this value as the *value of a potential partnership* between the endpoints. Nodes can participate in *at most one* exchange (i.e., they form *exclusive partnerships*); in other words, the partnerships form a matching in G. The outcome of such an exchange network is thus a matching in G, and assignment of values to players representing the players' share of the value of the partnership they (potentially) participate in:

Definition 9.1. An **exchange network** is a pair (G, v), where $G = (V, E)$ is a graph and $v : E \to \mathbb{N}$ is a function. An **outcome** of an exchange network (G, v) is a pair (M, a) where:

- M is a matching in G (i.e. a subset of E where no two edges share a common endpoint.)

- $a : V \to \mathbb{N}$ is an allocation of values to nodes such that for every $e = (u, v) \in M$, $a(u) + a(v) = v(e)$, and for every node v not part of an edge in M, $a(v) = 0$.

To see how this notion captures a matching market, consider a *bipartite* G—the "left nodes" correspond to buyers and the "right nodes" correspond to the items; the value of an edge e between a buyer i and an item y is simply the value v player i has for the item y. The exclusive partnership condition here models the fact that players only want one item, and each item y can only be sold to a single player. When the price p for an item y has been set, and the item gets sold to some buyer i, player i's value, v, for the item y gets split between i and the seller—p gets allocated to the seller and $v - p$ gets allocated to the buyer i.

But exchange networks can capture significantly more general strategic interactions, *even if G is bipartite*—we no longer need to view the nodes as "buyers" and "items/sellers", but rather just as individuals in a social network that engage in some form of exclusive partnerships. In particular, we can now see how some nodes in the social network are "more powerful" than others (due to the network structure), in the sense that they will get a "better deal" in the partnerships they establish. Indeed, such interactions have been empirically studied in the sociological field of "network exchange theory" [Wil99].

To do this, we first need to have a way to generalize the notion of market equilibria to exchange networks. The notion of a *stable outcome* does this.

9.2 Stable Outcomes

We call an outcome (M, a) *stable* if no node x can improve its allocated amount by establishing a *new partnership* with some node y such that a) x is *strictly better off* than in its current allocation, and b) y receives *at least as much* as it does in its current allocation. Intuitively, if such a pair of nodes (x, y) exists (i.e., the outcome is **unstable**), then we would expect x to break its current partnership and propose a new partnership to y which y would accept.[1] A mathematically cleaner way of saying the same thing is to require that for every pair of nodes (x, y), their currently allocated *combined* amount, $a(x) + a(y)$, is strictly smaller than the value of their potential partnership, $v(x, y)$: if $a(x) + a(y) < v(x, y)$, then x and y can, for instance, form a new partnership

[1] If amounts are infinitely divisible, x can always offer y a deal that strictly increases also y's allocated amount (by some small ϵ), while still ensuring that x is strictly better off—this is why we expect y to also want to break its current partnership.

with the allocation $a'(x) = a(x) + s/2$ and $a'(y) = a(y) + s/2$ (one of them rounded upwards and the other downwards), where $s = v(x, y) - a(x) - a(y)$ is the *surplus* on the (x, y) edge—since the surplus is positive, at least one of them will be better off and the other at least as well off.

Definition 9.2. Given an exchange network (G, v), we call an outcome (M, a) **unstable** if there exists a pair of nodes x, y in G such that $(x, y) \notin M$, but $a(x) + a(y) < v(x, y)$ (i.e., the edge has a positive surplus). If (M, a) is not unstable, we call it **stable**.

In other words, an outcome is stable if and only if every edge has a surplus of 0. Let us now show that that this notion indeed generalizes that of a market equilibrium. First, we need to formalize the correspondence between matching markets and exchange networks:

- Given a matching market $\Gamma = (n, Y, v)$, we say that (G, v') is the **exchange network corresponding to** Γ if: a) $G = (V, E)$ where $V = [n] \cup Y$ and $E = \{(u, u') : u \in [n], u' \in Y\}$ (i.e. G is the complete bipartite graph over the given nodes), and b) $v'(i, y) = v_i(y)$.
- Given a exchange network (G, v') where $G = (([n] \cup Y), E)$, is a bipartite graph, we say that $\Gamma = (n, Y, v)$ is the **matching market corresponding to** (G, v') where $v_i(y) = 0$ if $(i, y) \notin E$, and $v_i(y) = v'(i, y)$ otherwise.

We now have the following claim.

Claim 9.3. *Let $\Gamma = (n, Y, v)$ be a matching market and let (G, v) be the exchange network corresponding to Γ. Or, conversely, let (G, v) be an exchange network where $G = ([n \cup Y], E)$ is bipartite, and let Γ be the matching market corresponding to it. Then (p, M) is a market equilibrium in Γ if and only if (M, a) is a stable outcome in (G, v) where:*

- $a(y) = p(y)$ *if* $y \in Y$
- $a(i) = v_i(M(i)) - p(M(i))$ *if* $i \in [n]$ *and* $M(i) \neq \perp$
- $a(i) = 0$ *otherwise.*

Proof. We separately prove each direction.

The "only-if" direction: Assume for contradiction that (p, M) is a market equilibrium in Γ, but (M, a) is not stable in (G, v')—that is, there exists some

unmatched buyer-item pair (i, y) such that $a(i) + a(y) < v(i, y)$. By our construction of a, this means that

$$v_i(M(i)) - p(M(i)) + p(y) < v_i(y)$$

Thus, $v_i(M(i)) - p(M(i)) < v_i(y) - p(y)$, which means that i would strictly prefer to buy item y at its current price (to buying item $M(i)$ that it currently is getting), which contradicts the market equilibrium property of (p, M).

The "if" direction: Conversely, assume for contradiction that (M, a) is stable in (G, v'), but (p, M) is not a market equilibrium in Γ—that is, there exists some buyer i that prefers buying item y at price $p(y)$ to the item (or lack thereof) $M(i)$ to which they currently are assigned. Assume first that i is matched to some object $M(i) \neq \perp$. Then, we have that

$$v_i(M(i)) - p(M(i)) < v_i(y) - p(y)$$

By the definition of a and v', the LHS equals $a(i)$ and the RHS equals $v'(i, y) - a(y)$; thus, we have

$$a(i) < v'(i, y) - a(y)$$

which contradicts the stability property of (M, a). Consider, finally, the case that is $M(i) = \perp$. Then $a(i) = 0$, but, since i prefers to buy y, $v_i(y) > p(y) = a(y)$. So

$$a(i) + a(y) = a(y) < v_i(y) = v(i, y)$$

which, again, contradicts the stability property of (M, a). ■

Let us point out that, *a-priori*, there is a surprising aspect of Claim 9.3: the notion of a market equilibrium only says that no *buyer* wishes to switch to some other item/seller—it does not impose any restriction on the seller not wanting to switch to some other buyer (which may be willing to pay more)—yet the notion of stability (when considering the exchange network corresponding to a matching market) treats buyers and sellers symmetrically and requires that neither of them prefer switching to some other partnership. So, why are we getting this "no-deviation" property also for sellers (for free)? The point is that if a seller y could propose a partnership to buyer i that is better for y, that means there is a surplus on the (i, y) edge, which in turn means that i can also propose a partnership to y that i prefers to its current deal (i.e., there indeed exists some buyer i that prefers to switch to some object y that it currently is not matched with).

9.3 Existence of Stable Outcomes

A direct application of Claim 9.3, combined with Theorem 8.8 (which states that market equilibria always exists in a matching markets), gives an existence theorem for stable outcomes in bipartite graphs.

Theorem 9.4. *Any exchange network (G, v) where G is bipartite has a stable outcome. Furthermore, such a stable outcome can be found in time polynomial in the size of G and the maximum value of any edge in the graph.*

Proof. Observe that the notion of stability does not depend on the names/labels of the nodes in the graph, so we may without loss of generality assume that set of "left nodes" in the bipartite graph G are $[n]$. By Claim 9.3, a stable outcome in (G, v) exists if and only if there exists a market equilibrium in the matching market Γ corresponding to (G, v); by Theorem 8.8 such equilibria always exists. ■

A natural question is whether stable outcomes exists in *all* (and not just bipartite) graphs. The answer turns out to be no, even for very simple graphs.

Claim 9.5. *There exists an exchange network with three nodes for which no stable outcome can exist.*

Proof. Consider a "triangle" G with three nodes (i.e., all nodes are connected by an edge), and for every edge e, set the value of the edge, $v(e)$ to some integer $k \geq 2$. Assume, for contradiction, that we have a stable outcome (M, a) in G. Since G is a triangle, any matching in G can contain at most one edge. If M contains no edges, $a(x) = 0$ for every node x, and thus for any pair of edges (x, y), $a(x) + a(y) = 0 < v(x, y) = k$, so (M, a) is not stable. Consider next the case when M contains one edge (x, y). One of the nodes, say x, must have $a(x) \leq \frac{k}{2}$, and the third node z must have $a(z) = 0$ (since it is unmatched); thus $a(x) + a(z) \leq \frac{k}{2} < k$, which means the outcome is not stable. ■

Note that the triangle example shows that any outcome, in fact, must have some edge (x, z) with surplus at least $\frac{k}{2}$. So, if x, z split the surplus evenly (i.e., x offers himself $3k/4$ and offers $k/4$ to z), both players gain at least $k/4$ by forming this new partnership. This example thus rules out even more relaxed notions of stability where a deviation is only deemed viable if both player are *substantially* better off than in their current allocation.

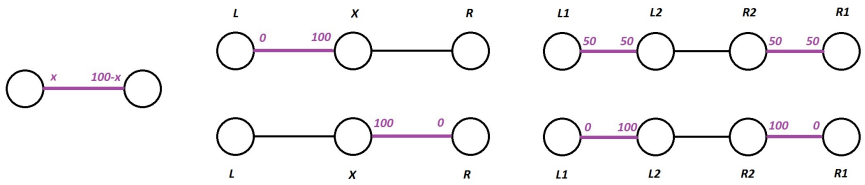


Figure 9.1: Some basic examples of exchange network games, with stable outcomes indicated in purple.

9.4 Applications of Stability

Let us now turn to applying the notion of stability to some example graphs. (All these graphs are bipartite, so we are guaranteed to have stable outcomes.) Consider the graphs in Figure 9.1, and let $v(e) = 100$ for every edge e; think of the value as 100 cents. (Several interesting behavioral experiments have been performed to understand exchange networks in laboratory setting. In these experiments, subjects are given some small example of a social network—such as the examples in Figure 9.1—and are each assigned to a node in the network, and then need to negotiate a split of a dollar with their neighbors, under to the exclusive partnership rule.)

- First, look at the leftmost example, consisting of two nodes with a single edge between them. Typically, we would expect the most "fair" outcome to be a 50/50 split of the dollar, but in fact any split is stable, as neither player is able to make a more advantageous deal with another player. (Usually, in behavioral experiments, we do see something close to a 50/50 split though; we will shortly return to this point.)

- Next, the second example, with three nodes in a line, has only two stable outcomes. The middle player X can choose to partner with L or R, but in any stable outcome *he gets all of the 100 units*. Otherwise, if X where to partner with, say, L and receives less than 100, then the (X, R) edge has a positive surplus! So we see that X here is "infinitely" powerful—due to its network position, it can extract out all the value!

- In the final, rightmost, example, first note that no stable outcome can contain a matching between the middle nodes (L_2, R_2); if it did, one of the nodes (w.l.o.g., L_2) would receive an allocation of at most 50, and there would thus be a surplus of at least 50 between this node and its

currently unmatched neighbor (i.e. L_1). From this, we conclude that any stable outcome must match L_1 to L_2, and R_1 to R_2. However, we cannot say much more than this: any allocations of the 100 units over these two edges with the property that L_2 and R_2 receive at least 100 combined (such as the ones illustrated in the figure where, in the first example, all players get 50, and, in the second example, L_2 and R_2 both take 100) will be stable.

Cut-off Thresholds: Rejecting Unfair Deals An important realization from these examples is that nodes on the "ends" of the lines—that is, nodes that are connected to only one other node—have basically no bargaining power; as they have no alternative options, they are forced to take whatever their single prospective partner offers them! However, in real life, even such isolated nodes have some small amount of power. To better understand the issue, consider the following simple game referred to as the **ultimatum game**:

- We have two players, A and B.
- A begins by proposing any split of one dollar between himself and B, as long as B is getting at least 1 cent.
- B can either *accept* the split and take the money he is assigned by A, or *reject* the split, in which case both A and B get nothing.

One might think that splitting the dollar 50/50 would be an equilibrium of this game. But in fact, it is not, since A can always deviate by increasing his cut of the split. In fact, the only PNE of this game is for *A to propose that he takes the whole pot (99 cents) and for B to accept* (since accepting is strictly dominant for B, and then A best-responds by taking the whole pot). This is clearly unrealistic—nobody would accept such an unfair deal in real life! Typically, there is some "cutoff threshold" under which an actual person would find the split to be offensive and refuse it, even if it means that they lose a small amount of money. (In fact, in behavioral experiments, this cut-off threshold has been found to be cultural and to depend on several other factors—for instance, a recent study shows that people typically have a higher cut-off threshold when drunk [MKA14].)

To capture this phenomena in the network exchange setting, we can, given some "actual" exchange network, consider (and analyze) an enlarged "mental-experiment" exchange network where each node gets assigned a new "moon" node x_m connected only to x, where the edge (x, x_m) has a value equal to x's cut-off threshold. In this moon-enlarged network, no node x would never

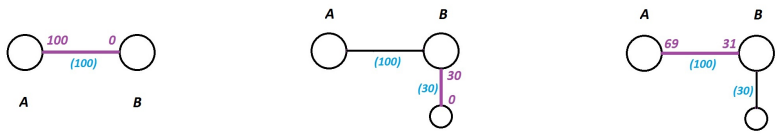


Figure 9.2: *Left*: An example of the "ultimatum game": A can propose an arbitrarily unfair split of the edge weight (100) to B in this exchange network. *Middle*: We can give B a "moon" node with edge weight 30 to denote their refusal to accept any deal where they receive less than 30. In this case, B will prefer to match with their "moon" instead of accepting A's deal from before. *Right*: However, if A proposes a favorable deal (in this case, 31), then B will still accept it.

accept an partnership where he earns less than $v(x, x_m)$, since it then it would just match with x_m.

9.5 Balanced Outcomes

As seen in even the simple two-node example, where essentially *any* outcome was stable, stability does not always provide enough predictive power. As such, it would be desirable to have some stronger notion of stability that provides more fine-grained predictions of what outcomes one should expect to see.

Nash Bargaining Let us first consider the simple example of a two-node graph. Here, we would expect to see an even split between the players.[2] This is arguably the "fair" outcome in this simply situation.

But how can we generalize this notion to more complicated graphs? To motivate the definition, consider again a simple two-node network with players A and B and where the value of edge between them is 1, but, now, let us also assume that A has some "outside option" such that he can choose to not match with B and still earn some utility x; similarly, let B have an outside

[2]This may not always be the case; for instance, in situations where one of the player inherently has more "social power" (e.g., the left node is the boss of the right node), we would expect to see an uneven split. Again, this phenomena can be captured by relying on the moon-modified network mentioned above, where the player with more inherent social power gets a "bigger moon".

option which guarantees him utility y. Let $s = 1 - x - y$ denote the surplus that the two players jointly earn from bargaining with one another instead of taking their outside options. Clearly, A will never accept any offer that gives him less than x, and B will never accept any offer where he earns less than y. So, effectively, they are now negotiating over how their surplus s should be split; effectively, this brings us back to the simple two-node example, where we would expect the surplus to be split evenly.

So A should receive $x + s/2$ (possibly rounded), and B should receive $y + s/2$ (possibly rounded); this amounts to a proper split of $v(x, y)$ that is similarly attractive (except for the possible rounding) to both players given their outside options. John Nash, in his paper on bargaining [Nas50a], indeed suggests that this is the "fair" way to set prices in such a bargaining situation.

Defining Balanced Outcomes We now rely on the Nash bargaining solution to obtain a strong notion of stability for general exchange networks; we say that an outcome is **balanced** if, for each edge in the matching, the allocation corresponds to the Nash bargaining solution wherein each node's "outside options" are defined by the allocations in the rest of the network—namely, the outside option of a node x is defined to be $\max_{y \neq M(x)}\{v(x, y) - a(y)\}$, that is, the maximum value it can earn by making an acceptable offer to some player besides the one it is currently matched with. Note that any balanced outcome clearly also is stable—each node clearly gets at least as much as its outside option which is what the notion of stability requires.

Revisiting the Examples We can now revisit the simple line networks considered in Figure 9.1; for each network, we now have a unique balanced outcome. In the leftmost graph, by definition, the balanced outcome is a 50/50 split. In the second graph, X still gets the full 100 units (as this was the only stable outcome, and as such can also be the only balanced outcome). The most interesting case is the rightmost graph. As argued before, there is not stable matching where the middle nodes L_2, R_2 get matched, and thus they cannot get matched in a balanced outcome either. Note, however, that the middle nodes still have the *option* of matching with each other and thus have an outside option, whereas the exterior nodes do not (i.e., their outside option is 0). Let l_1, l_2, r_2, r_1 denote the respective allocations in some balanced outcome. Since L_2's outside option is $100 - r_2$, and L_1 outside option is 0, by balance we have that

$$l_2 = 100 - r_2 + (100 - (100 - r_2))/2 = 100 - r_2/2$$

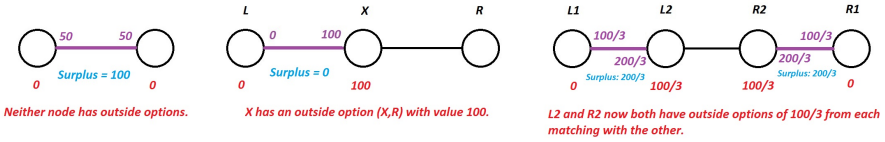


Figure 9.3: Returning to the examples in Figure 9.1, we can analyze the balanced states of each of these by considering each node's outside options. Outside options are indicated in red, surpluses in blue, and the balanced matchings and splits in purple.

By the same argument, we have that

$$r_2 = 100 - l_2/2$$

Solving this linear equation yields the solution $l_2 = r_2 = 200/3$; we conclude that $l_1 = r_1 = 100/3$. So, as expected, the interior nodes can leverage the fact that they have better "outside options" to get a better deal (i.e., $\frac{2}{3}$ of the 100 units, whereas the the exterior nodes are only getting $\frac{1}{3}$). See Figure 9.3 for a summary.

Balanced outcomes indeed provide good predictions of how human players will actually split money in behavioral experiments, not only in the case of a two-node network, but also in more general networks (such as the line networks considered). Additionally, a result by Kleinberg and Tardos [KT08] shows that every exchange network that has a stable outcome also has such an outcome that is balanced; this result, however, is outside the scope of this course.

Notes

Exchange networks for bipartite graphs were first considered by Shapley and Shubik [SS71]; the notion of stability is a special case of the notion of *the core* introduced by Gillies [Gil59] and studied for exchange networks in [SS71]. As far as we know, exchange networks in general graphs were first considered in [KT08]. The notion of a balanced outcome first appeared in [Roc84, CY92] but the term "balanced" was first coined in [KT08]; as mentioned, the Nash bargaining outcome was first studied in Nash's paper [Nas50a].

Theorem 9.4 was first proven in [SS71] and a generalized version of it appears in [KT08]; both these proofs rely on heavier machinery (solving linerar/dynamic programs). Our proof (relying on the connection between exchange networks and matching markets is new (as far as we know).

Finally, see [EK10, Wil99] for further discussion of behavioral experiments studying exchange networks.

Part III

Mechanisms for Networks

Chapter 10

Mechanism Design and Auctions

In our earlier study of matching markets, we demonstrated that market-clearing prices exist and that the market can adjust itself to reach them over time. We now turn to a "single-shot" version of this problem: we still have a set of buyers and a set of items for sale, but this time we consider a situation where the sale of the items is being administered by a central entity ("the auctioneer") as opposed to it happening in a decentralized way in a market.

We are interested in studying *mechanisms* that such an auctioneer can use to 1) incentivize players to *truthfully* report their values for the items, and 2) assign items to buyers in a way that *maximizes social value* (and thus, by Corollary 8.6, also social welfare). (As an aside, in many situations the auctioneer may also be the seller of the items (e.g., in the context of Google selling advertising slots); in such situations, the auctioneer may instead want to choose a mechanism that maximizes their own revenue rather than social welfare. This leads to a separate question that can be considered in the same framework. We here focus on "ideal" mechanisms adopted by a "benevolent" auctioneer and thus only consider social value maximization.)

To consider this question, and generalizations of it, let us begin by introducing a general framework for mechanism design.

10.1 The Mechanism Design Model

Consider a scenario where we have:

- A set of n players.

- A finite set of states Ω.
- $T = T_1 \times T_2 \times \ldots \times T_n$ is the product of sets T_i; we refer to T as the *type space*, and T_i as the *type space for player i*.
- Each player i is associated with a *valuation function* $v_i : T_i \times \Omega \to \mathbb{N}$, where $v_i(t_i, \omega)$ describes how a player i having type $t_i \in T_i$ values a state $\omega \in \Omega$.[1]

We refer to such a tuple (n, Ω, T, v) as a **mechanism design context**, or simply a **context**.

Let us now consider the following process induced by a mechanism $\mathcal{M}$:

- Each player i submits a "report" r_i to the mechanism $\mathcal{M}$. Ideally, the players submit their true type—that is, $r_i = t_i$— but players need not necessarily be "truthful".
- $M(\vec{r})$ outputs a state $\omega \in \Omega$ and a profile of prices/payments $\vec{p} \in \mathbb{R}^n$; we refer to the tuple $(\omega, \vec{p})$ as the **outcome**.
- The utility of a player i with type t_i in an outcome $(\omega, \vec{p})$ is then defined to be

$$u_i(t_i, \omega, \vec{p}) = v_i(t_i, \omega) - p_i$$

In other words, player i receives as utility its value of the outcome x minus the payment p_i it is charged by the mechanism. This type of a utility function is referred to as a *quasi-linear* utility function: note that it generalizes the utility function used in our study of matching markets. (As an advanced comment, let us point out that, while this way of defining utility is seemingly the most natural and simple way to capture the "happiness" of a player, there are some situations where the use of quasi-linear utilities are not appropriate—for instance, the fact that utility is linear in the payment p_i does not capture the phenomena that people treat a price gap of \$10 very differently depending whether they are buying, say, laundry detergent or a car. Nevertheless, in situations where the payments p_i are "small" compared to the wealth of the players, this model seems reasonable.)

To better understand the notion of a context and the above-described process, let us consider some basic examples, starting with the concept of an *auction*.

[1] It may seem redundant for players to be associated with both a type t_i as well as an *individualized* valuation function v_i. Indeed, formally, it suffices to consider a single valuation function v and assume that the type t'_i contains information about who the player is (e.g., let $t'_i = (i, t_i)$), but this makes notation more cumbersome, and makes the notion of a type less intuitive.

Example: First- and Second-price auction. Consider the following setting:

- We have a set of n buyers, and a single item for sale.
- The states $\Omega = [n] \cup \perp$ determine which of the n players will receive the object (where $\perp$ represents no player winning).
- Player i's type $t_i \in T_i = \mathbb{N}$ describes i's valuation of the object, and $v_i(t_i, \omega) = t_i$ if $\omega = i$ (i.e., i gets the object) and 0 otherwise.[2]

We refer to a context $\Gamma = (n, \Omega, T, V)$ as above as a **single-item auction context**.

Now, consider the following **first-price auction mechanism** $\mathcal{M}$ for any such single-item auction context: The mechanism $\mathcal{M}$—run by the auctioneer, upon receiving reports (i.e., bids) r_i from each player i—returns $\mathcal{M}(\vec{r}) = (i^*, \vec{p})$, where:

- $i^* = \arg\max_i r_i$ (i.e. the winner is the player who reports (or "bids") the highest value); in the case of ties, some fixed *tie-breaking* procedure is used to select a winner among the highest bidders.
- $p_i = 0$ if $i \neq i^*$ (only the winner should pay for the item).
- $p_{i^*} = r_{i^*}$ (the winner should pay the amount that they bid).

We can similarly define a **second-price auction**, where the winner is still the highest bidder, but they instead need only pay the *second-highest bid*; that is,

$$p_{i^*} = \max_{j \in [n] \setminus i^*} r_j$$

10.2 Goals of Mechanism Design

The goal of mechanism design is to, given some context $\Gamma = (n, \Omega, T, v)$, design a mechanism $\mathcal{M}$ for this context which ensures that "rational play" leads to some "desirable" outcome, *no matter what types the players have.*

Desirability of Outcomes. As mentioned above, in our study of mechanism design, the notion of desirability will be to maximize social value (but as mentioned, other notions of desirability are also useful—for instance, in the

[2]Note that we here are making use of the fact that v_i is parametrized by the player identity i to determine whether i is actually receiving the object.

setting of an auction, we may consider the notion of maximizing the seller's revenue.) Given a context Γ, let the social value be given by

$$SV^{\Gamma}(\vec{t}, x) = \sum_{i \in [n]} v_i(t_i, x)$$

Given a context $\Gamma = (n, \Omega, T, v)$ and a type profile $\vec{t} \in T$, we say that a state x **maximizes social value** if

$$x = \arg\max_{x \in X} SV^{\Gamma}(\vec{t}, x).$$

We can also define social welfare as we did in Chapter 8 by additionally incorporating the prices paid by the players as well as the profit made by the mechanism operator (e.g., the "seller" in the setting of an auction); by the same argument as in Claim 8.5, these prices will cancel out with the mechanism operator's profit, and so social welfare will be equal to social value.

Truthful Implementation Let us now turn to defining what we mean by "rational play". Several interpretations of this are possible. Ideally, we would like to have a mechanism where "rational" players will *truthfully report their types*. To formalize this using concepts from game theory, we must first view the process as a game: Given a context $\Gamma = (n, \Omega, T, v)$ a type profile $\vec{t} \in T$, and a mechanism $\mathcal{M}$, let $\mathcal{G}^{\Gamma, \vec{t}, M}$ denote the game induced by $\Gamma, \vec{t}$, and $\mathcal{M}$, where each player i chooses some report r_i (as their action) and their utility is defined as above, based on the state and prices ultimately chosen by $\mathcal{M}$.

We now have the following natural notion of what is means for a mechanism to be truthful.

Definition 10.1. A mechanism $\mathcal{M}$ is **dominant-strategy truthful (DST)** for the context $\Gamma = (n, \Omega, T, v)$ if, for every $\vec{t} \in T$, t_i is a dominant strategy for player i in $\mathcal{G}^{\Gamma, \vec{t}, M}$.

So, if M is DST, then players are incentivized to report their true type (e.g., for auctions, their true valuation of the object) *regardless of what other players choose to do*! This is a relatively strong notion; we may also consider a seemingly weaker notion, which simply requires that the action profile where all players truthfully report their types is a Nash equilibrium:

Definition 10.2. A mechanism $\mathcal{M}$ is **Nash truthful** (NT) for the context $\Gamma = (n, \Omega, T, v)$ if, for every $\vec{t} \in T$, $\vec{t}$ is a Nash equilibrium in $\mathcal{G}^{\Gamma, \vec{t}, M}$.

As it turns out, these notions are equivalent:

Claim 10.3. *A mechanism $\mathcal{M}$ is DST if and only if it is NT.*

Proof. If M is DST, then it is clearly NT. Conversely, let us assume for the sake of contradiction that M is NT and not DST; that is, there exist some types $\vec{t}$, some player i and reports r_{-i}, such that t_i is not a best-response w.r.t r_{-i} assuming players have the types $\vec{t}$. We then claim that t_i is not a best-response w.r.t r_{-i} assuming players have the types (t_i, r_{-i})—this directly follows from the fact that the utility function of player i only depends on i's valuation and payment and is independent of other players' types. It follows that M is not NT since i wants to deviate given the types $\vec{t'} = (t_i, r_{-i})$, which is a contradiction. ■

Given the notion of DST, we can now define what it means to for a mechanism to implement social value maximization.

Definition 10.4. A mechanism $\mathcal{M}$ is said to **DST-implement social value maximization** for the context $\Gamma = (n, \Omega, T, v)$ if M is DST for Γ and, for every $\vec{t} \in T$, $M(\vec{t})$ maximizes social value with respect to Γ and $\vec{t}$.

Nash Implementation. We may also consider a weaker notion of implementation, which we refer to as *Nash-implementation* (in contrast to *Nash-truthful implementation*), which only requires the existence of *some* Nash equilibrium (not necessarily a truthful one) that leads to a social-value-maximizing outcome:

Definition 10.5. A mechanism $\mathcal{M}$ is said to **Nash-implement** social value maximization for the context $\Gamma = (n, \Omega, T, v)$ if, for every $\vec{t} \in T$, there exists a PNE $\vec{t'}$ for $\mathcal{G}^{\Gamma, \vec{t}, M}$ such that $\mathcal{M}(\vec{t'})$ maximizes social value with respect to Γ and $\vec{t}$.

Revisiting the First- and Second-Price Auctions. Let us now examine whether the examples of first-price and second-price auctions satisfy our desiderata. Clearly, the simple first-price auction is not truthful; if everyone else values an object at 0, and you value it at 100, you would clearly be better off bidding something less than your actual value (say, 1), thereby saving a lot of money buying the item! Generally, bidding your true value always provides a utility of 0 (since you pay what you bid), so underbidding can never be worse. However, for the second-price auction, we have the following beautiful result.

Theorem 10.6. *The second-price auction DST-implements social value maximization.*

Proof. We start by showing that the second-price auction is DST. By Claim 10.3, it actually suffices to show that it is NT. Consider some player i with valuation v, and let v^* be the highest valuation of all the players, *excluding* i; that is:

$$v^* = \max_{j \neq i} v_j$$

First, note that by bidding v, player i can never end up with negative utility—either the bid is losing (and he get 0 utility), or he gets the item for the second highest bid, which is at most v. We next argue that i cannot improve their utility by either overbidding or underbidding, by considering the following cases:

Case 1: $v < v^*$**:** As long as i bids below v^* (or at v^* while losing the tie-breaking), his utility remains 0. In contrast, if he bids above v^* (or at v^* while winning the tie-breaking), his utility becomes negative—he needs to pay v^* for an item that he values v).

Case 2: $v > v^*$**:** As long as i bids above v^* (or at v^* while winning the tie-breaking), his utility remains unchanged—he still needs to pay v^*. In contrast, if i bids below v^* (or at v^* while losing the tie-breaking), his utility becomes 0.

Case 3: $v = v^*$**:** If i looses the auction his utility is 0, and if he wins, he needs to pay $v^* = v$, so again his utility is 0.

We conclude that the second-price auction is NT and thus also DST. Finally, by construction, if everyone bids truthfully, then the player (or one of the players, in the case of ties) who values the item the most will win the auction, thus the auction chooses an outcome that maximizes social value. ■

First-price auctions, while not truthful, still have a useful property:

Theorem 10.7. *The first-price auction mechanism Nash-implements social value maximization.*

Proof. Given any type (valuation) profile $\vec{t}$, let i^* be the player with the highest valuation (following any tie-breaking procedures implemented). Let i^{**} be the player with the second-highest valuation. Consider the bids $r_i = t_i$ if $i \neq i^*$ and $r_{i^*} = t_{i^{**}}$ (i.e., the player with the highest valuation, bids the second highest valuation, and everyone else bids their true valuation). Players besides i^* can only receive negative utility by deviating from reporting their true valuation. Meanwhile, i^* (the winner of the item) loses utility by overbidding,

and receives 0 (losing the item) by underbidding. So $\vec{r}$ is a PNE. Additionally, since the object is still assigned to the player who values it the most, the auction also implements social value maximization. (Perhaps surprisingly, notice that in this Nash equilibrium, the pricing is identical to a second-price auction!) ■

Nash-implementation v.s. DST implementation Notice that in the case of a first-price auction, in order for the players to figure out what the Nash equilibrium strategy actually is, they need to *know each others' valuations*! While this may be reasonable to assume in a world where the auction is run repeatedly (among the same set of players), it would be difficult to believe that this equilibrium could occur in a single-shot auction. Instead, it seems more likely that players would "shade" their bids (i.e., underbid) based on their *beliefs about the other players' valuations*. (Formalizing this requires assumptions about the distributions of players' valuations; we briefly discuss this setting in the notes at the end of the chapter.)

In contrast, if we have a DST mechanism, everyone knows hows to bid, independent of their beliefs about the other players valuations (and strategies); indeed, this is why DST implementations are so desirable.

10.3 The VCG Mechanism

A natural question following the above discussion is whether we can find a mechanism that implements social-value maximization in *any context* (and not just for single-good auctions). In fact, the Vickrey-Clarke-Groves (VCG) mechanism shows that this is possible; in particular, as we shall later see, we can use this mechanism to design an auction mechanism for matching markets.

Theorem 10.8. *For every context Γ, there exists a mechanism $\mathcal{M}$ that DST-implements social value maximization for Γ.*

Proof. Given a context $\Gamma = (n, \Omega, T, v)$, the mechanism $\mathcal{M}(\vec{r})$—which we refer to as the **plain VCG** mechanism—outputs an outcome $(\omega^*, \vec{p})$ such that:

- $\omega^* = \arg\max_{\omega \in \Omega} \mathrm{SV}^{\Gamma}(\omega, \vec{r})$ (i.e., let ω^* be the state that maximizes social value, $\mathrm{SV}^{\Gamma}(\cdot, \vec{r})$, given reports $\vec{r}$)
- $p_i = -\sum_{j \neq i} v_j(r_j, \omega^*)$. (This is negative, so the operator will pay players rather than charging them.)

If every player truthfully reports their type, then by the definition of ω^*, this mechanism will select an outcome that maximizes social value. Let us now

argue that M is DST. Consider some player i with type t_i; assume the other players submit r_{-i}. Then, if the mechanism selects the outcome $(\omega^*, \vec{p})$, player i obtains utility equal to

$$v_i(t_i, \omega^*) - p_i = v_i(t_i, \omega^*) + \sum_{j \neq i} v_j(r_j, \omega^*) = SV^{\Gamma}(\omega^*, (t_i, r_{-i}))$$

So, player i should submit a report r_i such that M chooses ω^* to maximize this expression. But, by submitting t_i, M will, by construction, pick such a state. So, submitting $r_i = t_i$ is a dominant strategy; hence M is DST and DST-implements social value maximization. ■

Computational efficiency of the Plain VCG mechanism. Note that if we can efficiently find the socially optimal outcome, then the Plain VCG mechanism becomes computationally efficient.

The Clarke pivot rule: paying your "externality". In the Plain VCG mechanism described above, the operator has to pay the players, which often is not desirable. We can modify the mechanism so that the players instead need to pay the operator. Specifically, if we shift the price of player i in a manner that is independent of i's report, the resulting mechanism will remain DST, since the price adjustment does not affect i's preference between actions; that is, we now let,

$$p_i = V_i(r_{-i}) - \sum_{j \neq i} v_j(r_j, \omega^*)$$

for any function $V_i(\cdot)$. We refer to any such mechanism (obtained by shifting the prices in the Plain VCG mechanism by V_i) as a **VCG mechanism**. A particularly useful instance—referred to as the **Clarke-pivot rule**—is specified by letting

$$V_i(r_{-i}) = \max_{\omega \in \Omega} \sum_{j \neq i} v_j(r_j, \omega)$$

That is, we are adjusting the prices by the "maximum social value if we were to exclude player i from the game". (As such, just as for the "plain VCG mechanism", the VCG mechanim with the Clark pivot rule is computationally efficient if we can efficiently find outcomes maximizing social value.)

Thus, player i now needs to pay:

$$p_i = \max_{\omega \in \Omega} \sum_{j \neq i} v_j(r_j, \omega) - \sum_{j \neq i} v_j(r_j, \omega^*)$$

Note that this is always a non-negative amount, since the social value excluding i at the state ω^* can clearly never be higher than the maximum social value excluding i.

A natural interpretation of the VCG mechanism with the Clarke pivot rule is that each player i pays for the "harm" in value he causes the other players by being present in the game; this is called the player's *externality*. With i present, the other players get a total of $\sum_{j \neq i} v_j(r_j, \omega^*)$; without i present, they would get $\max_{\omega \in \Omega} \sum_{j \neq i} v_j(r_j, \omega)$. In other words,

$$p_i = [max_\omega SV_{-i}(\vec{r}, \omega)] - SV_{-i}(\vec{r}, \omega^*)$$

where $SV_i(\vec{t}, \omega)$ denotes the social value of all players except i, assuming that the players types are $\vec{t}$, and where ω^* is the outcome that maximizes social welfare for everyone *including* i.

Let us look at two brief examples of a VCG mechanism.

Example: Airport or Train station A small town needs to decide whether to build an airport or a train station; let $\Omega = \{A, T\}$ (A for deciding to buld the airport and T for building the train station.) For every inhabitant of the town (i.e., each player) i, let the type space T_i be the set of function from Ω to $\mathbb{N}$, representing the monetary value they have for each outcome, and let the valuation function for each player i be $v_i(t, \omega) = t(\omega)$.

For instance, consider a situation where we only have 2 players, and let $t_1(A) = 10, t_1(T) = 0$ (i.e., player 1 strongly needs an airport and but has no need for a train station), and let $t_2(A) = 4$, $t_2(T) = 6$ (i.e., player 2 weakly prefers a train station) . Consider applying the VCG mechanism with the Clarke pivot rule to this context, assuming the players have types t_1, t_2, and they truthfully report their types to the mechanism (as proven above, doing so is a dominant strategy). Clearly, the outcome, ω^*, which maximizes social value is A (with a social value of $10 + 4 = 14$ as opposed to $0 + 6 = 6$ for the outcome T), thus the VCG mechanism will pick this outcome assuming players truthfully report their types.

Now, consider the externality of each player. Without player 1 present, T would be the best outcome, and player 2 would receive 6 in utility; however, in the outcome ω^*, player 2 receives only 4, thus player 1's externality is 2. Player 2's externality is 0, since the outcome is the same regardless of whether player 2 is present or not. Thus, player 1 would be charged a price of 2, whereas player 2 would not be charged anything.

Example: single-item auctions (recovering the second-price auction). Consider applying the VCG mechanism with the Clarke pivot rule

for single-item auction contexts. Clearly, the item will be assigned to the player i^* with the highest bid (as this maximizes social value). The externality of player i^* is equal to the second-highest bid: if i^* were not present, the second-highest bidder i^{**} would win and receive value equal to their bid, but if i^* is present, i^{**} gets nothing. None of the other players' value is affected by i^* showin up or not (they still do not get it), so the externality of player i^* is exactly the second-highest bid. (Note that it is important that we consider the *value* and not *utility* for players when computing the externality!) No other players have externalities, since their presence or absence does not affect the outcome (i^* will win either way). So they pay nothing. Hence, VCG with the Clarke pivot rule gives us exactly a second-price auction when applied to single-item auction contexts!

10.4 VCG and Matching Markets

Let us now apply VCG to the problem of designing a matching market auction. Recall that a matching market (n, Y, v) specifies a set $[n]$ of players, a set Y of objects, and for each player i, a valuation function $v_i : Y \to \mathbb{N}$. Such a matching market frame directly corresponds to a **matching market context** (n, Ω, T, v):

- Ω (the set of states) is the set of all possible allocations $a : [n] \to Y \cap \perp$ such that for $i \neq j$, either $a(i) \neq a(j)$ or $a(i) = a(j) = \perp$ (i.e., either two players are assigned different items, or they both are unassigned).
- For each i, T_i is the set of all possible valuation functions $v : Y \to \mathbb{N}$.
- $v_i(t, a) = t(a(i))$ if $a(i) \neq \perp$ and 0 otherwise.

We can now directly use VCG mechanisms on the social-choice context to get a matching market auction which DST-implements social value maximization (and hence social welfare maximization). See Figures 10.1 and 10.2 for an illustration of both the Plain VCG and VCG with the Clarke pivot rule in the context of matching markets. For concreteness, the VCG mechanism with the Clarke pivot rule proceeds as follows:

- Pick the allocation a^* that maximizes $\sum_{i \in [n]} r_i(a^*(i))$.
- Set prices $p_i = \max_{a \in X} \sum_{j \neq i} r_j(a(j)) - \sum_{j \neq i} r_j(a^*(j))$ (i.e. p_i is i's externality).

Computational Efficiency: Notice that to efficiently implement this mechanism, we need a way to efficiently find the matching (i.e., allocation a^*) that

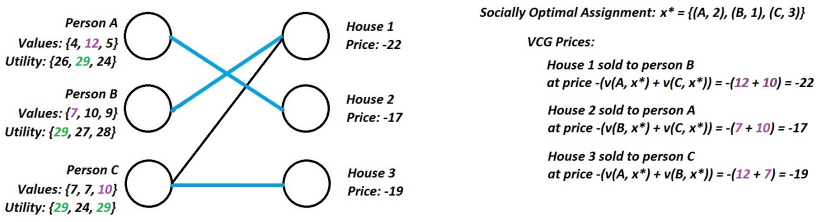


Figure 10.1: An illustration of "plain" VCG pricing without the Clarke pivot rule. The price each player "pays" is the negative of the other players' social value in the socially optimal equilibrium.

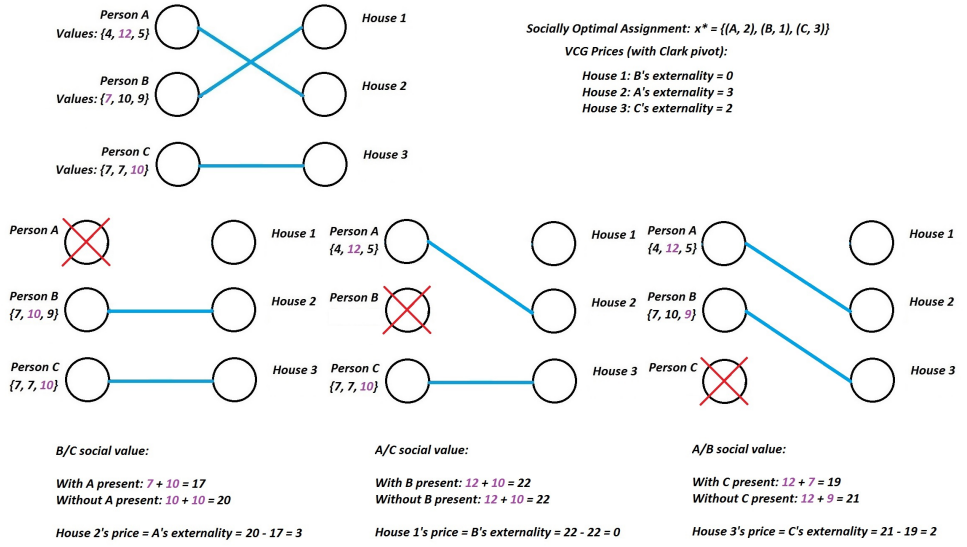


Figure 10.2: An illustration of VCG pricing using the Clarke pivot rule. The price each player pays is their externality, or the amount by which their presence in the auction decreases the other players' social value. Notice that the Clarke VCG price for a player is equal to the standard VCG price plus the other players' social value with that player absent.

maximizes social value. If we think of $v_i(t_i, y)$ as being the weight of edge (t_i, y) in a bipartite graph, this amounts to finding the *maximum-weight bipartite matching* of this graph. There are various direct ways of doing this; here, we simply remark that a combination of the market-clearing theorems we have already demonstrated shows how to efficiently find such a matching that maximizes social value. Namely, Theorem 8.8 gives us a polynomial-time algorithm for finding a market equilibrium, which by Corollary 8.6 maximizes social value.

Revisiting bundles of identical goods. Let us return to the case where we have a matching market in which the items are bundles of identical goods; recall that bundle i contains c_i goods, and we assume without loss of generality that $c_1 > c_2 > ...c_n$. A frame for this type of matching market again corresponds to a social choice context (n, Ω, T, v):

- Ω (the set of states) is the set of all possible allocations $a : [n] \to [n]$ such that $a(i) \neq a(j)$ if $i \neq j$.
- For each i, $T_i = \mathbb{N}$—that is, each player's type is now simply an integer (its value per individual object).
- $v_i(t, a) = c_{a(i)}t$.

We will refer to such a context as a **market-for-bundles context**. Let us again consider the VCG mechanism with the Clarke pivot rule for this context. Then $M(\vec{r})$ proceeds as follows:

- Pick an allocation a^* that maximizes social value (i.e. $\sum_{i\in[n]} c_{a^*(i)}r_j$); for concreteness, let $k(i)$ be the i**'th ranked player**—that is, the player with the i'th highest valuation, breaking ties lexicographically—and assigning the i'th bundle to player $k(i)$.
- Then, we need to charge the i'th ranked player, $k(i)$, its externality. Assuming truthful reporting, the optimal allocation for players ranked j such that $j < i$ does not change if we remove player $k(i)$, so it suffices to consider the change in value for players ranked $j > i$. If player $k(i)$ is present, these players receive total value

$$\sum_{j>i} c_j r_{k(j)}.$$

  But if player i is absent, each of these players gets the next-larger bundle, for a total value of

$$\sum_{j>i} c_{j-1} r_{k(j)}.$$

So, the i'th ranked player's externality (and thus the price charged to them) is the difference between these values:

$$p_{k(i)} = \sum_{j>i} (c_{j-1} - c_j) r_{k(j)}$$

Note that a seemingly curious property of this auction is that the last ranked player gets the smallest bundle for free. This seems unrealistic. It turns out that this is not a real issue: If the seller also has some value for the items, as already discussed in Chapter 8, we could add an "extra buyer" representing the seller's interests (i.e,. the extra buyer's value would be the seller's value for an item). If the seller's value is lowest ones, this values will serve as a **reserve price**, and the lowest ranked "actual" buyer will always have to pay at least this reserve price.

10.5 Generalized Second-Price (GSP) Auctions

While the VCG pricing for matching markets for bundles is relatively simple to compute, the mechanism is still a lot more complicated than the "simple" first and second price auctions for the case of single-item auction. As we have seen, in the case of a single-item auction, VCG (with the Clarke pivot rule) actually reduces down to a second-price auction. A natural question is whether we can get a similarly simple mechanism also for multi-item auctions. We restrict our attention to markets for bundles context.

Consider, now, the following natural generalization of the second-price mechanism: assign the i^{th} bundle to the i^{th} ranked player, $k(i)$, and let him pay the $(i+1)^{\text{st}}$ ranked bid *per item*—that is,

$$p_i = c_i r_{k(i+1)}$$

if $i < n$ and $p_n = c_n r_{k(n)}$. This mechanism, which was introduced by Google, is referred to as the *generalized second-price (GSP)* mechanism.

GSP is not DST Despite the fact that the GSP seems like a natural generalization of the second-price auction (which we know is DST for the case of single-item auctions), GSP is not DST when considering multiple bundles! To see this, consider the following example: let us say we have three bundles of items with $c_1 = 10$, $c_2 = 4$, $c_3 = 0$ and three players with unit values $t_1 = 7$, $t_2 = 6$, $t_3 = 1$. If the players bid truthfully, player 1 gets the largest bundle (10) for the second-highest price (6), thus earning utility $7(10) - 6(10) = 10$.

But if player 1 were to deviate and report his valuation at 5, he would get the second-largest bundle (4) at the third-highest price (1), earning utility $7(4) - 1(4) = 24$. So in this case bidding truthfully is not a PNE!

GSP Nash-implements social-value maximization However, despite the fact that GSP is not truthful, it is actually the mechanism that Google and other sellers of advertising slots use for sponsored search advertisements! Among other factors, this may be because the simplicity of the auction is appealing—bidders can easily understand how much they will have to pay. In addition, as we now show, GSP does Nash-implement social value maximization:

Theorem 10.9. *The GSP mechanism Nash-implements social value maximization in any market-for-bundles context.*

Proof. It suffices to show that there is some Nash equilibrium in the induced game that maximizes social value. Assume without loss of generality that player k has the k^{th} ranked valuation for the items being sold. By Theorem 8.8, there exists a market equilibrium in the matching market corresponding to this social choice context. By Theorem 8.7, this equilibrium maximizes social value, and so we can assume, without loss of generality, that in this equilibrium player k is assigned bundle k (since we ordered players by decreasing valuation and bundles by decreasing size; if a tie causes players to be assigned bundles out of order, we can reorder them accordingly while still maintaining the equilibrium property).

Now, let α_i be the price per item in bundle i; by Theorem 8.11, we have that $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n$. Consider the bidding strategy where $r_i = \alpha_{i-1}$ if $i > 1$ and $r_1 = \alpha_1$. Since $r_k \geq r_{k+1}$ for any k, and because we have assumed ties to be broken lexicographically, the GSP mechanism will assign player k to bundle k, and so $\vec{r}$ maximizes social value by the above. It remains to show that $\vec{r}$ is a Nash equilibrium.

Notice that, by construction of $\vec{r}$, player i will receive the same bundle as in the market equilibrium (i.e., bundle i) and *at the market-clearing price* α_i. Let us now argue that player i has no incentives to deviate. Consider some deviation by player i. If the deviation leads to the same assignment for i, its utility remains the same, so the deviation is not profitable. Let us thus consider some deviation where i gets assigned a different bundle j. We distinguish two cases:

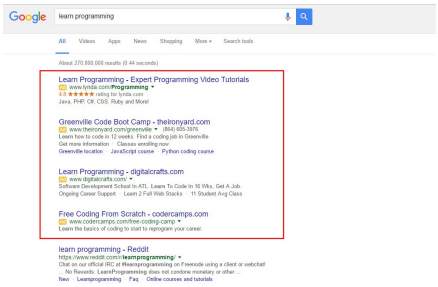


Figure 10.3: An example of sponsored search in action. Notice that of the displayed results here, the first four are advertisements; these sponsored search slots are sold to advertisers, with the most prominent (higher) slots being worth the most.

- **Overbidding:** If i overbids so that it now gets assigned a *bigger* bundle $j < i$, it "pushes down" player j (to position $j+1$) and will thus have to pay the bid $r_j \geq \alpha_j$ of player j per item.
- **Underbidding:** If i underbids so that it now gets assigned a *smaller* bundle $j < i$, it "pushes up" player j (to position $j-1$) and will thus have to pay the bid $r_{j+1} = \alpha_j$ of player $j+1$ per item (just as player j previously did).

Thus, in both cases, i gets assigned a bundle j and has to pay at least α_j for it. By the market clearing property of $\vec{\alpha}$, player i must prefer its current bundle i at price α_j to getting bundle j at price α_j, so in neither case, the deviation can be profitable for i. ■

10.6 Applications to Sponsored Search

An important real-life example of a matching market auction is the market for sponsored search. Here, a search engine (e.g., Google) sells advertising "slots" for search terms. For instance, when a user makes a search for "learn programming", Google has four advertisements that appear before the actual search results; see Figure 10.3. To determine which ads to show, Google uses an auction for bundles of identical goods. The items being sold are "clicks" (i.e. instances of a user clicking on the ad), and the "size" of each of the bundles (slots) is the number of clicks an advertiser would expect to get from each of them—the *clickthrough rate*. (We are incorrectly assuming that clickthrough

rates are the same no matter who the advertiser is; this will be discussed shortly.) Each advertisers i's, type t_i is its value for a click.

In such a scenario, we can use either the VCG mechanism or the GSP mechanism to sell these advertising slots, and charge the buyers accordingly. Our previous analysis applies directly if we charge the advertiser for each search taking place—that is, we charge per *impression* of the ad—the (expected) value for advertiser i for a slot j with click-through rate c_j is $c_j t_i$, just as in a market-for-bundles context.

Charging per click. Search engines typically no longer charge per impression of an ad, but instead charge by the *click*. The auctions can easily be modified to capture this setting by instead charging a user i placed on slot j p_i/c_j per click; in the case of GSP, this simplifies the auction even further, as we can now simply charge the i^{th} ranked user the $(i+1)^{\text{st}}$ ranked bid!

Dealing with ad quality. So far in this discussion, we have assumed that the clickthrough rate per slot is independent of the advertiser (or ad) being placed there. This is clearly unrealistic—a relevant ad, for instance, is far more likely to get more clicks than an irrelevant "spam" advertisement. To model this, we assume that each advertiser i has its "quality" or discount parameter q_i—an "optimal advertiser" has $q_i = 1$, but a spammer may have a significantly lower value of q_i—and an advertisement i placed on slot j is assumed to get $c_j q_i$ clicks. The quality q_i of an advertiser i is estimated by the search engine, and we thus assume that the seller is aware of it.

Our previous treatment almost directly extends to deal with this. First, note that we can directly apply it if we simply think of c_j as the "potential" number of clicks received by an "optimal advertiser", and if we think of the type t'_i of a player i as its value for a "potential click"—namely, $t'_i = t_i q_i$, where t_i is i's actual value for a click. It is more appealing, however, to keep player i's type t_i as its actual value for a click, and charge them for each "real" (as opposed to potential) click. That is easily achieved as follows:

- As before, each player i submits a bid r_i for a click.
- The mechanism first computes a "discounted bid" $r'_i = r_i q_i$ (i.e., if r_i had been the true value of a click, r'_i will be the true value for a "potential click").
- We next run the VCG or GSP mechanisms for bundles assuming player i's bid is r'_i.

- Finally, let

$$p'_i = \frac{p_i}{q_i c_{a(j)}}$$

and output $a, \vec{p}'$. (Recall that p_i is the price for the impression; thus, $\frac{p_i}{c_{a(j)}}$ is the price for a "potential click", and $\frac{p_i}{q_i c_{a(j)}}$ becomes the price for an actual click.)

Notes

The development of mechanism design theory began with the work of Hurwicz [Hur60]. The VCG mechanism was introduced and analyzed in the works of Vickery [Vic61], Clarke, [Cla71] and Groves [Gro73]. The generalized second-price auction was introduced and analyzed in [EOS07].

Let us also mention that the literature on mechanism design often also considers a setting, referred to as *Bayesian*, where players' types come from a probability distribution D (typically, we assume that player types are independent). In such a setting we may then consider a weaker notion of a "Bayes-Nash truthful" mechanism, where every player's *expected* utility (where the expectation is over the distribution of other players' types) is maximized by truthfully reporting their type; this is formalized by defining a generalized version of a normal-form game called a *Bayesian Game* where players first receive a type sampled from D and then need to choose their action (without knowing the actual type that was sampled for the other players.) The notion of a Bayes-Nash truthful implementation is no longer equivalent to DST; optimal Bayes-Nash truthful auctions were studied in work of Myerson [Mye81]. Leonid Hurwicz, Eric S. Maskin and Roger B. Myerson received the Nobel prize in Economics (for laying the foundation of mechanism design theory) in 2007.

As mentioned above, the VCG mechanism is computationally efficient as long as we have a computationally efficient method for finding socially optimal outcomes. For many optimization problems, however, it is *computationally hard* to find the socially optimal state (even if people truthfully report the preferences); consequently, the computer science literature has focused on studying, so-called, *approximation algorithms*, where we contend ourselves with finding a "close-to-optimal" outcome. Can we simply plug in such an approximation algorithm into the VCG mechanism? It turns out that the resulting mechanism no longer is DST. In fact, (assuming the existence of "secure encryption schemes",) *no computationally efficient* "general-purpose" mechanisms (like VCG)—that simply use some underlying approximation algorithm

as a "black-box"—can be DST [PS14, CIL12]. Consequently, the computer science literature has focused on directly designing DST mechanism that attempt that guarantee "close-to-otimal" outcomes: this study—referred to as *algorithmic mechanism design*—was initiated by Nisan and Ronen [NR99]. See [NRTV07] for a detailed study of this area.

Chapter 11

Voting: Basic Notions

In this chapter, we consider the question of designing mechanisms for voting (i.e., electing a candidate). In the context of social networks, this is useful e.g., for electing a leader/representative for a set of users. In subsequent chapters, we will additionally apply many of the ideas developed in the context of voting various types of matching markets, and web search.

To phrase the task of voting as a mechanism design problem, we first need to define the mechanism design context.

11.1 Social Choice Contexts

Consider a mechanism design context (n, Ω, T, v) where:

- the outcome space Ω is a set of candidates X from which we wish to elect a single winner,
- for each player i, the type space T_i for i is a set of preferences over the candidates X; formally, a **preference** $\mu \in T_i$ is a ranking (i.e., an ordering) $\mu = (x_1, \ldots, x_{|X|})$ of the elements in X; to simplify notation, we let $x >_\mu x'$ denote that x is higher ranked than x' according to μ (and define $<_\mu$ analogously).
- for every player i, $t \in T_i$, $v_i(\mu, x) \geq v_i(\mu, x')$ if $x >_\mu x'$)

We refer to such a context as a **social-choice context**; if it always holds that $v_i(t, x) > v_i(t, x')$ (i.e., we have strict inequality) when x is higher ranked than x' according to t, we refer to the context as a **social-choice context with strict preferences**. Furthermore, if for each player i, T_i is the full set of preferences over candidates X (i.e., all orderings are possible), we refer to the context as a **complete social-choice context**.

For concreteness, consider a simple (complete) social-choice context with two players (voters) (i.e., $n = 2$), two candidates, $X = \{A, B\}$, voter 1's preferences are (A, B) (i.e., he prefers A to B), whereas voter 2's preferences are (B, A), and for both players $i \in [2]$, $v_i(t, x) = 10$ if x is the highest ranked outcome according to t and 0 otherwise.

As we saw in Chapter 10, we can use the VCG mechanism to DST-implement social value maximization for any context, and thus, in particular for any social-choice context—such a mechanism ensures the selection of the candidate that maximizes social value assuming players act rationally (we revisit the VCG mechanism in the context of voting in Section 12.3).

But, recall that the VCG mechanism requires using payments. Typically, when considering *voting rules* (for e.g., presidential elections), it is desirable to have a scheme without any payments (as the use of payments may e.g., unfairly prioritize the votes of people with more money to spend.)[1] Consequently, we here explore mechanisms for selecting "desirable outcomes" *without payments*.

11.2 Voting Rules and Strategy-proofness

We refer to a mechanism $\mathcal{V}$ without payments (i.e. one which will always outputs prices $\vec{p} = \vec{0}$) as a **payment-free mechanism**. A **voting rule** $\mathcal{V}$ is simply a payment-free mechanism for a social-choice context.

To simplify notation, when discussing payment-free mechanism $\mathcal{V}$, we simply ignore the payment part of the output of $\mathcal{V}$ and simply write $\mathcal{V}(\vec{\mu}) = x$ to mean that $\mathcal{V}$ selected the outcome x.

We will be interested in voting rules that incentivize players to truthfully report their preferences over candidates:

Definition 11.1. We say that a voting rule $\mathcal{V}$ is **strategy-proof** if $\mathcal{V}$ is DST.

That is, if $\mathcal{V}$ is strategy-proof there does not exists some "situation" (formally, a profile of preferences for all the players) such that some rational player will want to vote "strategically", lying about their preferences to influence the outcome—hence the name "strategy-proofness".

[1] As a side-remark, it can be argued that systems like the American election system do not fully satisfy this no-payment condition, as donations to parties or candidates may be viewed as a way to use money to influence the voters' preferences and as a consequence also the outcome of the election; in our treatment, we ignore these effects—our formal model assumes that the type t_i of a player i, which determines i's preferences among candidates, is fixed and cannot be influenced by others.

11.3 Condorcet Voting

As we shall see later on, even in very simple social-choice contexts, we cannot hope to reproduce the success of the VCG mechanism without using payments: there are no voting rules that DST-implement social value maximization. But even in situations where we can, it is not clear that this is actually a reasonable desiderata in the context of voting: any social-value maximizing mechanism must put a bigger weight on voters who are more strongly affected by the outcome of the election—for instance, if the utility function represents financial gains, the vote of a billionaire (who risks to loose a lot if e.g., the tax code is changed) needs to count significantly more than the vote people with more modest income/assets.

A natural desideratum would instead be to elect a candidate that *a majority of the voters prefer to all others*:

Definition 11.2. Given a set of voter preferences $\mu_1, \dots, \mu_n$ over a set of candidates $X = \{x_1, \dots, x_m\}$, we say that a candidate $x \in X$ is a **Condorcet winner** if for every other $x' \in X$ at least $n/2$ voters prefer x to x' (according to the preferences $\vec{\mu}$).

When there are only two choices, the most obvious voting rule works:

Theorem 11.3. *For every social-choice context Γ over two candidates, there exists a voting rule $\mathcal{V}$ that is strategy-proof for Γ; furthermore, $\mathcal{V}$ will always output a Condorcet winner.*

Proof. Let $\mathcal{V}$ be the **majority voting rule**: simply pick the candidate that the majority of voters rank first, and break ties in some arbitrary way (e.g., lexicographically).[2] By definition, the candidate x elected by $\mathcal{V}$ is preferred to the other candidate by at least $n/2$ of the voters, and is thus a Condorcet winner.

Furthermore, no matter how other voters vote, a voter i can never improve their utility by not reporting its favorite candidate; the only time when voter i's vote would matter is in the event that there is a draw between the two candidates (or when i's vote would produce a draw instead of a loss for their candidate), and in these cases, i would never prefer voting for the other candidate to their own. ■

So, we have an excellent algorithm for n voters and two candidates. However, there are often more than two candidates. The obvious extension would

[2] Note that if exactly $n/2$ voters prefer each candidate, then *both* are Condorcet winners).

be a generalized version of the majority voting rule, called the **plurality voting rule**, where we consider only voters' top choices and select whichever candidate receives the most "top-choice votes" (this is, for instance, how American elections typically work at the state level), and as before, we break ties in some arbitrary but fixed way. If we have two candidates, then the plurality voting rule is equivalent to the majority voting rule.

11.4 The Problem with Non-binary Elections

So, with more than two candidates, does plurality output a Condorcet winner? Is it strategy-proof? Perhaps surprisingly, *neither* of these are true. As we first observe, with three or more candidates, *no voting rule can always elect a Condorcet winner!*.

Theorem 11.4. *Consider some complete social-choice context $\Gamma = (n, \Omega, T, v)$ with strict preferences where n is a multiple of three and $|\Omega| \geq 3$. There is no voting rule $\mathcal{V}$ for Γ that always outputs a Condorcet winner.*

Proof. Consider a complete social-choice context with 3 voters and three candidates A, B, C and consider a situation where voter's preferences are described as follows:

1. $A > B > C$
2. $B > C > A$
3. $C > A > B$

So 2/3 of the voters prefer A to B, 2/3 of the voters prefer B to C, and 2/3 of there voters prefer C to A. That is, for every candidate x, there exists some candidate x' such that a majority of the voters (more precisely, 2/3 of the voters) prefer x' to x. Thus, there does not exist a Condorcet winner (i.e., a candidate x that is preferred to every other candidate by half the voters), so no mechanism $\mathcal{V}$ can output one. The same proof obviously works even if the number of voters n is a multiple of 3 or if we add (dummy) candidates. ■

We turn to considering strategy-proofness:

Claim 11.5. *The plurality voting rule is not strategy-proof for any complete social-choice context $\Gamma = (n, \Omega, T, v)$ with strict preferences where $n \geq 3$ is an odd integer[3] and $|\Omega| \geq 3$.*

[3]The restriction on n is there only to simplify the proof; the claim is true assuming just that $n \geq 2$. We shall later see a much more powerful result (Theorem 12.2) which implies this claim as a special case, thus we here content ourselves with the weaker statement.

Proof. Consider a social-choice context with $n = 2k + 1$ voters and three candidates A, B, C; if the candidate space is larger, we simply ignore the other candidates. Consider a situation where voters' preferences are described as follows:

- k voters prefer $A > B > C$
- k voters prefer $B > A > C$
- and the final voter, prefers $C > B > A$

If everyone votes truthfully, either A or B will be selected depending on how tie-breaking is implemented. Let's first assume that A is selected. If so, the "final voter" who prefers C should clearly "lie" and cast e.g., $B > A > C$ as his vote—this ensures that B wins, which is a better outcome for him (according to his real preferences). If instead ties are broken so that B wins, the player who prefers C would instead prefer to lie if his preferences had been $C > A > B$. ■

Note that plurality only consider players' top-choices; we may also consider an even more generalized version of majority voting called a **scoring-based voting rule**: such a voting rule is specified by "scores/weights" w_j for each rank position j such that $w_1 \geq w_2 \geq ...$; the final score of a candidate is defined to be the sum of the "weights" it receives from the voters, and the candidate with the highest score is declared the winner. For instance, if $w_1 = 10$, $w_2 = 5$ and $w_j = 0$ where $j > 2$, a candidate's score is 10 times the number of voters that rank him first, plus 5 times the number of voters that place him second. Note that plurality is a special case of a scoring-based voting rule where $w_1 = 1$ and $w_j = 0$ if $j > 1$.

Can some scoring-based voting rule be strategy proof? The answer again is no. In fact, the Gibbard-Satterthwaite theorem shows that if we restrict to voting rules that are onto (i.e., all candidates can win), then the only strategy-proof voting rule over three or more candidates is dictatorship! We will explore this result in the next chapter.

Notes

The area of social choice theory was initiated in the papers by Condorcet [dC94] and Arrow [Arr50]. Condorcet also introduced the desiderate which today is called "Condorcet voting" and presented the impossibility result for condorcet voting.

Chapter 12

Voting: Barriers and Ways around Them

In this chapter, we continue our exploration of voting: we present a fundamental barrier to strategy-proof voting (the Gibbard-Satterthwaite Theorem), and next exemplify some ways to circumvent this barrier.

12.1 The Gibbard-Satterthwaite Theorem

To formally state this theorem, we will restrict to **onto** voting rules: rules where, for every candidate $x \in X$, there exists some profile of preferences/rankings $\vec{\mu}$ such that $V(\vec{\mu}) = x$. That is, it must be feasible for every candidate to win.

Let us now define what it means for a voting rule to be dictatorial.

Definition 12.1. Given a complete social-choice context $\Gamma = (n, \Omega = X, T, v)$, we say that a voter i is a **dictator** for $x \in X$ with respect to $\mathcal{V}$ if $\mathcal{V}(\vec{\mu}) = x$ for every $\vec{\mu}$ such that i ranks x first. We call a voting rule $\mathcal{V}$ **dictatorial** if there exists some voter i such that, for every $x \in X$, i is a dictator for x with respect to $\mathcal{V}$. (That is, the outcome of the election depends only on i's preferred outcome.)

A useful property of the notion of dictatorship is that if a voter i is a dictator for some candidate a, then no other voter i' can be a dictator for some *other* candidate $b \neq a$ (since the situation where i ranks a first and j ranks b first would lead to a conflict).

Note that any dictatorial voting rule $\mathcal{V}$ clearly is onto and strategy-proof. The Gibbard-Satterthwaite Theorem shows an unexpected converse of this observation:

Theorem 12.2 (The Gibbard-Satterthwaite Theorem)**.** *Let $\mathcal{V}$ be an voting rule that is onto and strategy-proof for some complete social choice context with strict preferences over at least three outcomes. Then $\mathcal{V}$ is a dictatorial.*

Note that a direct corollary of this theorem is that voting rules for social-choice contexts with strict preferences, in general, cannot hope to DST-implement social-value maximization (since for most social choice contexts, dictatorship clearly will not implement social-value maximization; we leave it as an exercise to prove this).

Proof of the Gibbard-Satterthwaite Theorem [Advanced]

For simplicity of notation, in the remainder of this section, we always have some fixed complete social-choice context Γ over at least three outcomes and with strict preferences, in mind (and as such, whenever we write strategy-proof, or onto, we mean with respect to this social-choice context). Before turning to the actual proof, let us first demonstrate that strategy-proof voting rules satisfy two natural properties that we would expect from any "reasonable" voting rule.

Definition 12.3. A voting rule $\mathcal{V}$ is **monotone** if for every two preference profiles $\vec{\mu}$ and $\vec{\mu}'$, if $\mathcal{V}(\vec{\mu}) = x$, and for every player i and every candidate y such that $x >_{\mu_i} y$ it is also true that $x >_{\mu'_i} y$, it follows that $\mathcal{V}(\vec{\mu}') = x$.

In other words, if a voting rule is monotone, then a winning candidate x will continue to win as long as, for any player i, all candidates that are dominated by x continue being dominated by x.

Definition 12.4. A voting rule $\mathcal{V}$ is **Pareto-optimal** if for every profile $\vec{\mu}$ and candidate x such that $x >_{\mu_i} y$ for every player i, we have that $\mathcal{V}(\vec{\mu}) \neq y$.

In other words, if a voting rule is Pareto-optimal, then no candidate y, that is dominated by some candidate x in every player's preference ordering, can ever win. In particular, if a candidate x is preferred by all players, that candidate must win.

The following two lemmas state that strategy-proof voting rules are both monotone and Pareto-optimal.

Lemma 12.5. *Any strategy-proof voting rule $\mathcal{V}$ is monotone.*

Proof. Assume for contradiction that $\mathcal{V}$ is not monotone; then, there exist profiles of preference $\vec{\mu}$ and $\vec{\mu}'$ so that $\mathcal{V}(\vec{\mu}) = x$ and for every player i and

	μ = μ0	μ1	μ2	μ3	μ4	μ5 = μ'
Voter 1:	1 > 2 > 3 > 4	-> 1 > 3 > 2 > 4	1 > 3 > 2 > 4	1 > 3 > 2 > 4	1 > 3 > 2 > 4	1 > 3 > 2 > 4
Voter 2:	1 > 4 > 2 > 3	1 > 4 > 2 > 3	-> 1 > 2 > 3 > 4	1 > 2 > 3 > 4	1 > 2 > 3 > 4	1 > 2 > 3 > 4
Voter 3:	2 > 3 > 4 > 1	2 > 3 > 4 > 1	2 > 3 > 4 > 1	-> 3 > 1 > 4 > 2	3 > 1 > 4 > 2	3 > 1 > 4 > 2
Voter 4:	3 > 4 > 1 > 2	3 > 4 > 1 > 2	3 > 4 > 1 > 2	3 > 4 > 1 > 2	-> 3 > 1 > 4 > 2	3 > 1 > 4 > 2
Voter 5:	4 > 3 > 1 > 2	4 > 3 > 1 > 2	4 > 3 > 1 > 2	4 > 3 > 1 > 2	4 > 3 > 1 > 2 ->	3 > 1 > 2 > 4
Winner:	1	1	1	3	3	3

Figure 12.1: An example of the hybrid preference profiles in the proof of Lemma 12.5. In this case, we can see that the winner does change between $\vec{\mu}$ and $\vec{\mu}'$, despite the fact that every player who prefers 1 to any other candidate in $\vec{\mu}$ still prefers 1 to the same candidates in $\vec{\mu}'$. The example shows that the voting rule (here, plurality with ties broken by largest candidate number) is not monotone and hence not strategy-proof.

every candidate y such that $x >_{\mu_i} y$ it is also true that $x >_{\mu'_i} y$, and yet $\mathcal{V}(\vec{\mu}') \neq x$.

We construct a sequence of "hybrid" preferences $\vec{\mu}^k$ where we move from $\vec{\mu}$ to $\vec{\mu}'$ one player at a time:

- $\mu_i^k = \mu_i$ if $i > k$, and
- $\mu_i^k = \mu'_i$ if $i \leq k$.

(See Figure 12.1 for an illustration of the hybrid preference profiles). Thus, $\vec{\mu}^0 = \vec{\mu}$ and $\vec{\mu}^n = \vec{\mu}'$. Now let k^* denote the smallest index where $\mathcal{V}(\vec{\mu}^{k^*}) = x$ but $\mathcal{V}(\vec{\mu}^{k^*+1}) = z \neq x$. Clearly such an index must exist, since $\mathcal{V}(\vec{\mu}^n) \neq x$ by assumption.

Furthermore, note that the only difference between $\vec{\mu}^{k^*}$ and $\vec{\mu}^{k^*+1}$ is that player $k^* + 1$ is switching from μ_{k^*+1} to μ'_{k^*+1}. Since $\mathcal{V}$ is strategy-proof, we must have that $x >_{\mu_{k^*+1}} z$, or else player $k^* + 1$ would prefer to report μ'_{k^*+1}, which would lead to the outcome z that they *strictly* prefer to x (by our assumption that preferences are strict). So, by the assumption on $\vec{\mu}'$, we have that $x >_{\mu'_{k^*+1}} z$ as well.

But, by the same argument, we have that $z >_{\mu'_{k^*+1}} x$ (or else player k^*+1 would prefer to report μ_{k^*+1}); this is a contradiction (since preferences are strict). ■

Lemma 12.6. *Any strategy-proof and onto voting rule $\mathcal{V}$ is Pareto-optimal.*

Proof. Assume for the sake of contradiction that there exists some preference profile $\vec{\mu}$ and some candidate x such that $x >_{\mu_i} y$ for every player i, yet $\mathcal{V}(\vec{\mu}) = y$. Now,

- Let $\vec{\mu}'$ be a modified version of $\vec{\mu}$ where candidates x, y are "pushed-up" so that x is every player's first choice and y is every player's second choice. *(For instance, if we have two voters with preferences $a > b > x > c > y$ and $c > x > a > y > b$ in $\vec{\mu}$, respectively, their preferences in $\vec{\mu}'$ will become $x > y > a > b > c$ and $x > y > c > a > b$.)* By monotonicity, we must still have that $\mathcal{V}(\vec{\mu}') = y$ (since all candidates dominated by y still are dominated by it).
- Since $\mathcal{V}$ is onto, there must exist some preference profile $\vec{\nu}$ such that $\mathcal{V}(\vec{\nu}) = x$. Let $\vec{\nu}'$ be the modified version of $\vec{\nu}$ where x and y are "pushed-up" in the same way as before). Again, by monotonicity, we have $\mathcal{V}(\vec{\nu}') = x$.
- Finally, by monotonicity applied to the transition from $\vec{\nu}'$ to $\vec{\mu}'$, it follows that $\mathcal{V}(\vec{\mu}') = x$, since we are only moving around candidates that are dominated by x (and y)). This is a contradiction.

■

We are now ready to prove the Gibbard-Satterthwaite Theorem.

Proof of the Gibbard-Satterthwaite Theorem. First, note that the theorem for the case that $n = 1$ directly follows from Pareto-optimality of $\mathcal{V}$. We next prove the theorem for the case that $n = 2$, and then proceed to prove the general case by induction.

The special-case of 2 voters ($n = 2$): Towards proving the theorem for the special-case of 2 votes, let us first prove the following easier claim.

Claim 12.7. *Consider some strategy-proof and onto voting rule $\mathcal{V}$ for two voters, and an ordered pair of candidates (a, b). Then, either player 1 is a dictator for a w.r.t. $\mathcal{V}$, or player 2 is a dictator for b w.r.t. $\mathcal{V}$.*

Proof. Consider any two candidates a and b. Let profile μ_1 be some arbitrary ranking where a is first and b is second, and let μ_2 be the same ranking except that we switch the order of a and b. *(For instance, if we let μ_1 be $a > b > x > y > z$; then μ_2 becomes $b > a > x > y > z$.)* Note that by Pareto-optimality either a or b must win the election (i.e. $\mathcal{V}(\vec{\mu}) \in \{a, b\}$), since all other candidates are dominated by both a and b for both players. Let us first consider the case that a wins (i.e., $\mathcal{V}(\vec{\mu}) = a$).

Now consider a new profile of rankings (μ_1, μ_2') where μ_2' is identical to μ_2 except a is ranked last for player 2. *(For instance, in the example above, μ_2' becomes $b > x > y > z > a$.)*

- By Pareto-optimality, we still have that either a or b must win, since all other candidates are still dominated by b.
- However, by strategy-proofness, b cannot win now, since in this case player 2 would have preferred to report μ_2' in the original profile $\vec{\mu}$ (where a wins but player 2 prefers b).
- So, a must still be winning with respect to the new profile (μ_1, μ_2').
- By monotonicity (transitioning from (μ_1, μ_2') to $\vec{\mu}''$) it follows that $\mathcal{V}(\vec{\mu}'') = a$ for *every* profile of rankings $\vec{\mu}''$ where player 1 ranks a on top, since in any such ranking a still dominates all other candidates for player 1 (and monotonicity holds vacuously for player 2, since a dominates no other candidate in μ_2').

Thus, we conclude that player 1 is a dictator for a with respect to $\mathcal{V}$. Symmetrically, in the case that $\mathcal{V}(\vec{\mu}) = b$, player 2 is a dictator for b. ■

Now we can apply this claim to three candidates.

Claim 12.8. *Given any set of candidates a, b, c, we have that either player 1 or player 2 is a dictator for all three candidates.*

Proof. We simply apply Claim 12.7 to different subsets of the three candidates:

- Applying Claim 12.7 to (a, b), we get that either 1 is a dictator for a or 2 is a dictator for b. Assume, for now, that the first case holds (i.e., 1 is a dictator for a).
- Next, applying Claim 12.7 to (b, c), we get that either 1 is a dictator for b or 2 is a dictator for c. But since we already assumed that 1 is a dictator for a, 2 cannot be a dictator for c (or else we would get a conflict in the definition of $\mathcal{V}$, as noted above). Hence 1 must be a dictator for both a and b.
- Finally, applying Claim 12.7 to (c, a), we get that either 1 is a dictator for c or 2 is a dictator for a. Again, 2 cannot be dictator for a since we have shown that 1 is a dictator b, thus 1 must also be a dictator for c.
- We thus conclude that 1 is a dictator for all three candidates. By symmetry, if the second case held initially (i.e., player 2 was a dictator for b), player 2 would become a dictator for all candidates.

■

We can finally conclude the proof of the theorem for the case that $n = 2$ by applying Claim 12.8 to all triples (a, b, x), $x \neq a, b$, and again observing that since 1 and 2 cannot simultaneously be dictators for both a and b, one of the players must be a dictator for every candidate.

The general case ($n \geq 2$). Next, to prove the theorem for any number of voters, we proceed by induction. The base case ($n = 2$) has already been proven. For the induction step, assume that the theorem is true for $n - 1$ voters; we shall prove it holds for n voters. Assume for contradiction that this is not the case. That is, there exists some onto and strategy-proof voting rule $\mathcal{V}$ with n voters that is not dictatorial.

Consider the voting rule $\mathcal{V}^x$ obtained by fixing the first player's vote to the preference profile where x is ranked highest and everyone else is ranked according to a lexicographic order. First, note that since $\mathcal{V}$ is strategy proof, we have that for every candidate x, $\mathcal{V}^x$ is strategy proof. Next, as we show below, $\mathcal{V}^x$ also is onto. (Looking forward, we will then use the induction hypothesis to conclude that $\mathcal{V}^x$—which has $n - 1$ voters—is dictatorial, and then use this fact to conclude that also $\mathcal{V}$ must be dictatorial.)

Claim 12.9. *$\mathcal{V}^x$ is onto for every $x \in X$.*

Proof. Towards showing this claim, consider a different voting rule $\tilde{\mathcal{V}}$ for just 2 voters:

$$\tilde{\mathcal{V}}(\mu_1, \mu_2) = \mathcal{V}(\mu_1, \mu_2, \mu_2, \ldots, \mu_2)$$

By $\mathcal{V}$'s Pareto-optimality, $\tilde{\mathcal{V}}$ must be onto, since if everyone prefers x then $\mathcal{V}$ must output x. Since $\mathcal{V}$ is strategy-proof, $\tilde{\mathcal{V}}$ must be strategy-proof for player 1; by the following argument, it must also be so for player 2:

- Assume for contradiction that $\tilde{\mathcal{V}}$ is not strategy-proof for player 2; then there exists μ_1, μ_2, μ_2' such that $\tilde{\mathcal{V}}(\mu_1, \mu_2') >_{\mu_2} \tilde{\mathcal{V}}(\mu_1, \mu_2)$.
- Now, consider $\tilde{\mathcal{V}}(\mu_1, \mu_2) = \mathcal{V}(\mu_1, \mu_2, \mu_2, \ldots, \mu_2)$ and switch one player at a time from μ_2 to μ_2' until we arrive at $\tilde{\mathcal{V}}(\mu_1, \mu_2')$; let $\vec{\mu}^k$ denote the kth such "hybrid" preference profile.
- Since, $\tilde{\mathcal{V}}(\mu_1, \mu_2') = \mathcal{V}(\vec{\mu}^n) >_{\mu_2} \tilde{\mathcal{V}}(\mu_1, \mu_2) = \mathcal{V}(\vec{\mu}^0)$, there must exist some k^* such that $\mathcal{V}(\vec{\mu}^{k^*+1}) >_{\mu_2} \mathcal{V}(\vec{\mu}^{k^*})$ and thus also $\mathcal{V}(\vec{\mu}^{k^*+1}) >_{\mu_{k^*}^{k^*}} \mathcal{V}(\vec{\mu}^{k^*})$ which contradicts strategy-proofness for player k^* w.r.t. $\mathcal{V}$.

So (by the already proven base case) $\tilde{\mathcal{V}}$ must be dictatorial. If player 1 is the dictator for $\tilde{\mathcal{V}}$, then, by monotonicity, it follows that 1 must be a dictator also for $\mathcal{V}$, which is a contradiction. Thus, player 2 must be the dictator for $\tilde{\mathcal{V}}$; but this implies that $\mathcal{V}^x$ is onto. ■

Thus, we have shown that for every x, $\mathcal{V}^x$ is both onto and strategy-proof. By the inductive hypothesis it thus follows that for every $x \in X$, $\mathcal{V}^x$ (having $n-1$ participating voters) is dictatorial. This does not directly give us that also $\mathcal{V}$ is dictatorial (which would conclude the proof) since *a-priori*, there may exists $x_1, x_2 \in X$, such that $\mathcal{V}^{x_1}$ has a *different dictator* (say i_1) than $\mathcal{V}^{x_2}$ has (say i_2). However, as we now show, if that were to happen, $\mathcal{V}$ could not be strategy-proof for player 1.

Consider a preference profile $\vec{\mu}$ where player i_1 ranks x_2 highest (and the rest lexicographically), and all other players rank x_1 highest (and the rest lexicographically). If everyone votes truthfully, since player 1 ranks x_1 highest, the outcome will be $\mathcal{V}(\vec{\mu}) = \mathcal{V}^{x_1}(\mu_{-1}) = x_2$ since i_1 is a dictator for $\mathcal{V}^{x_1}$. If player 1, however, had misreported his preferences and placed x_2 highest, then the outcome would be $\mathcal{V}^{x_2}(\mu_{-1}) = x_1$ since i_2 is a dictator $\mathcal{V}^{x_2}$. So player 1 prefers to misreport his preferences (ranking x_2 highest) and thus $\mathcal{V}$ is not strategy-proof, which is a contradiction.

We conclude that there exists a single player i such that for every $x \in X$, player i is a dictator for $\mathcal{V}^x$, and thus player i is a dictator also for $\mathcal{V}$; this concludes the proof. ■

12.2 Single-Peaked Preferences and the Median Voter Theorem

The above impossibility result relies on the fact that voters' preferences over candidates can be arbitrary. In other words, it relies on the fact that the social-choice context is *complete*. Under a natural restriction on the preferences, it can be overcome. In fact, we can also overcome the impossibility of Condorcet voting.

Roughly speaking, we say that preferences are *single-peaked* if all candidates can be placed on a *one-dimensional spectrum* (e.g. think of the American political spectrum in terms of liberal/left and conservative/right), and voters' valuations for candidates strictly decrease based on how far they are to the left or right of their "ideal outcome" on the spectrum—that is, the valuation decreases with distance to some ideal point, but might decrease differently depending on whether it is to the left or right of that point. See Figure 12.2 for an illustration.

More formally, we say that a social-choice context $\Gamma = (n, X, T, v)$ has **single-peaked preferences** if a) Γ has strict preferences, and b) there exists some "global" ordering $x_1, x_2, \ldots, x_n$ over the candidates in X such that for

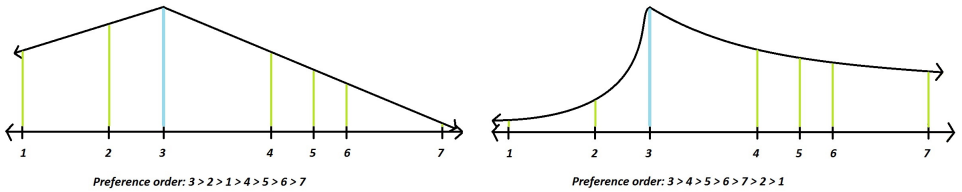


Figure 12.2: An illustration of the intuition behind single-peaked preferences. In this example, seven candidates are arranged on a spectrum. We can imagine the voter's preferences as a function over this spectrum; the only requirements for this function are that it must be maximized at the voter's top choice (in this case, 3), and strictly decreasing as distance from that choice increases. In the first example (left), the preference diminishes directly with distance; in the second example (right), it does not, which preserves the voter's top choice but changes the order of preferences. In fact, any preference where $3 > 2 > 1$ and $3 > 4 > 5 > 6 > 7$ is admissible as single-peaked.

each player i, the type set T_i is the set of preferences μ for which there exists some "ideal candidate" i^* such that for all $i < j$:

- if $i^* < i < j$, then $x_i >_\mu x_j$;
- if $i < j < i^*$, then $x_j >_\mu x_i$.

The celebrated *Median-Voter Theorem* by Black [Bla48] shows that once we restrict to social-choice contexts with single-peaked preferences, a Condorcet winner always exists—in fact, the preferred choice of the so-called "median voter" will be a Condorcet winner. We additionally remark that this median-voter mechanism is also strategy-proof:

Theorem 12.10 (The Median Voter Mechanism)**.** *There exists a strategy-proof voting rule $\mathcal{V}$ for all single-peaked social choice contexts; furthermore, $\mathcal{V}$ will always select a Condorcet winner for all single-peaked social choice contexts.*

Proof. For simplicity, let us begin by assuming an odd number of voters. Let $a_1, \ldots, a_n$ denote the top choices of each voter's reported preferences, ordered according to the one-dimensional spectrum. Let $\mathcal{V}$ output $a^* = a_{\frac{n+1}{2}}$ (i.e. the preference of the median voter according to the spectrum). See Figure 12.3 for an example.

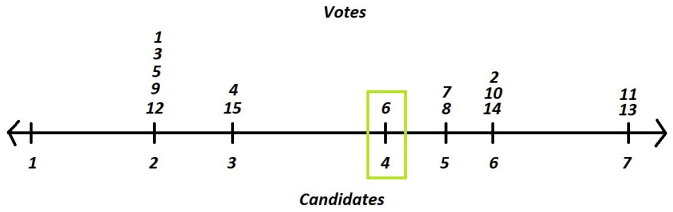


Figure 12.3: An example of the median voter rule with 15 voters and the seven candidates from the previous figure. If we order the fifteen votes according to their positions on the spectrum, voter 6 is the median (eighth) vote, and so his candidate (4) wins, despite candidate 2 having a plurality of the votes. This rule is strategy-proof assuming single-peaked preferences, but may still leave a lot of voters unhappy, and definitely does not maximize social welfare (*Exercise:* Try to come up with a preference function for each voter to show this!).

$\mathcal{V}$ produces a Condorcet winner. Consider any alternative candidate a_i such that $i < \frac{n+1}{2}$ (i.e. a_i is to the "left" of a^* on the spectrum). All $\frac{n+1}{2}$ voters including, and to the right of, a^* must prefer a^* to a_i, by the single-peaked preferences requirement. Symmetrically, if a_i is to the "right" of a^*, then all $\frac{n+1}{2}$ voters including, and to the left of, a^* must prefer a^* to a_i. So at least $n/2$ voters prefer a^* to any other candidate, and so a^* is a Condorcet winner.

$\mathcal{V}$ is strategy-proof. Clearly, the median voter is not incentivized to report his preference falsely, as they are receiving optimal utility (their top choice) by reporting truthfully. Voters to the left of the median cannot change the median by deviating, except by pushing it to the right by voting for a candidate to the right (and thus losing utility by the single-peaked preference rule); symmetrically, voters to the right can only push the median to the left by deviating (and again lose utility). Thus, the voting rule is strategy-proof.

Dealing with even n. Now, if n is even, we can just take the median vote to be $a^* = a_{n/2}$. We still have that $\mathcal{V}$ is strategy-proof by the same logic as above. In addition, $\mathcal{V}$ will still output a Condorcet winner; a^* will be preferred to an alternative candidate to the left by $n/2 + 1$ voters (all voters including

and to the right of a^*), and will be preferred to an alternative candidate to the right by at least $n/2$ voters (all voters including and to the left of a^*). ■

Notice that in case n is odd, the above proof shows that there is a unique Condorcet winner, whereas when n is even, there can be at most 2 ($a_{n/2}$ and $a_{n/2+1}$).

12.3 Voting with Payments

Finally, let us revisit the question of designing mechanisms for social-choice contexts, but this time allowing for payments—that is, the mechanism can now charge players based on how they vote. For concreteness, consider again a specific social-choice context with two players (voters) and where the set of outcomes X is $\{A, B\}$; recall that a mechanism M for this social-choice context needs to output a winning candidate *and* prices for the voters to pay. There are just two types of players $t_1 = (A, B)$, and $t_2 = (B, A)$; let the valuations for the players be such that player 1 (having type t_1) gets utility 10 if A gets elected, and 0 if B gets elected, whereas player 2 (having type t_2) get utility 4 if A gets elected and 6 if B.[1] (This example is essentially isomorphic to the "Airport v.s. Train-station station" example from Chapter 10, but using a different type space.)

Consider applying the VCG mechanism (with the Clarke pivot rule) to this context. The outcome maximizing social value is A winning (with a social value of 10+4=14 as opposed to 0+6=6 for B winning); thus, the VCG mechanism will pick this outcome assuming players truthfully report their types. Now, consider the externality of each player. If player 1 were not voting, B would win and player 2 would receive 6 in utility; however, with player 1 voting and A winning, player 2 only receives 4. Hence, player 1's externality is 2. Player 2's externality is 0, since the outcome is the same for player 1 regardless of whether player 2 votes or not. So in this case, player 1 would be charged a price of 2 by the VCG mechanism with the Clarke pivot rule, whereas player 2 would not be charged anything.

One interesting thing to note about this mechanism is that only a "pivotal player" who can actually change the outcome of the election has to pay anything (in the example above, player 1); they, in addition, never need to pay more than their difference in value between the two candidates.

[1] Formally, player 1's utility function is such that the player gets 10 in utility if its top choice wins (according to the player's type) and 0 otherwise, whereas player 2's utility, is such that it gets 6 if its top choice wins, and 4 otherwise.

Knowledge of Valuations Note that to apply the VCG mechanism to a social-choice context, the mechanism needs to know by how much each player values their respective candidate (i.e., the the mechanism needs to know the utility function)—this is needed since the players' types (that they report to the mechanism) are simply a ranked list of candidates. It is not clear how the voting authority (running the VCG mechanism) can obtain these valuations in a reliable way.

But there is an easy fix: just as we did in "Airport v.s. Train example" in Chapter 10, let us consider a more general mechanism design context for social choice where a player's type not only specifies a preference order, but also specifies how valuable each candidate is to the player! We can still apply the VCG mechanism to such a context, and the result would be that players now are incentivized to truthfully report exactly how valuable each candidate is to them (and we would still recover the same outcome described above, except that now player 1 would report $(A, 10), (B, 0)$ and player 2 would report $(A, 4), (B, 6)$).

So, we have a practical mechanism that enables us to elect a candidate that maximizes social value. Why aren't all the countries in the world switching to using this VCG mechanism? We highlight a few answers (think about others yourself!):

- As already mentioned, in the context of voting, maximizing social value may not be the "right" desiderata: a richer player may have more to gain/loose from the outcome of the election, and such a player's preferences will receive a heavier weight. This, intuitively, seems unfair and undesirable.
- But even if we decide that social value maximization is what we want to achieve, there is a problem with running this mechanism in an uncontrolled environment: *Collusion*. Consider a scenario with 100 voters, 98 of whom prefer A, but the other two of them want to bias the election so that B wins and they do not have to pay anything. Let us say that they both decide to bid an outrageous sum of money (say, 10 billion dollars), equal to far more than the combined sum of the A supporters valuations. Clearly, if either one of them were to not vote, the outcome of the election be B, so, *neither of them has any externality, and they would thus pay nothing!* This is an example of how collusion between players breaks the truthfulness of the VCG mechanism. (We mention that the same problem occurs also in the case of auctions, which is why there are regulations against collusion when running e.g., spectral auctions).

12.4 Summarizing the Voting Landscape

Let us informally summarize the key feasibility and infeasibility results obtained for strategy-proof voting:

- For the case of 2 candidates, there exists a strategy-proof voting rule that always selects a Condorcet winner: in fact, the most natural rule—*plurality*—of selecting the candidate with the most number of top choice votes works.
- For the case of 3 or more candidates, the only strategy-proof voting rule is dictatorship (as shown by the "Gibbard-Satterthwaite Theorem"); and no voting rule can always elect a Condorcet winner.
- But with restricted preferences (single-peaked ones, where the candidates can be placed on a one-dimensional spectrum), a natural voting rule (selecting the "median voter's top choice") is both strategy-proof and always selects a Condorcet winner.
- Finally, with payments, DST voting maximizing social value is possible (but the mechanism is vulnerable to attacks by collusion between two or more voters).

Notes

The Gibbard-Satterthwaite Theorem was first proved in [Gib73, Sat75]; this results is closely related to an earlier impossibility result by Kenneth Arrow [Arr50], demonstrating that a different set of desirable properties for voting rules are inconsistent; Arrow received the Nobel prize in Economics in 1972.

Our proof of the Gibbard-Satterthwaite Theorem is new but follows high-level ideas from the proofs in [SR14] and [BP90]. Single-peaked preferences and the "Median-Voter" Theorem are due to Black [Bla48].

The Gibbard-Satterthwaite theorem can be extended in various ways. First, the theorem only talks about *deterministic* voting rules, but as shown by Gibbard [Gib77] it can also be extended to randomized ones. Additionally, the notion of strategy-proofness is quite strong as it requires an implementation in dominant strategies (that is, truthfully reporting your preferences is always a best-response, no matter what the other players do). As mentioned in the previous chapters, a weaker form of truthful implementation is that of Bayes-Nash truthful implementation; in the context of voting, the notion of "Bayes-Nash strategy-proofness" would require that truthfully reporting your preferences maximizes expected utility assuming everyone else reports their true preferences, where the expected utility is computed assum-

ing all other player's types are independently drawn from some distribution D—this distribution D describes how players believe others will vote (e.g., D could be obtained from polling). Unfortunately, as shown by McLennan [McL11], the Gibbard-Satterthwaite theorem actually extends also to this setting! However, as shown in [LLP15], McLennan's result requires Bayes-Nash strategy-proofness to hold w.r.t. *all* distributions D, even those that are very "complicated" in the sense that they require high precision (i.e., a large number of decimal points) to describe the probabilities assigned to various candidates. In contrast, [LLP15] demonstrates natural voting rules that satisfy Bayes-Nash strategy-proofness w.r.t distributions D where only a small number of decimal points are used to describe probabilities (i.e., when players' beliefs are "coarse" as one would expect them to be.)

Chapter 13

Matching Markets without Money Transfers

In this chapter, mirroring our treatment of voting, we revisit the matching market problem in a setting without money transfers.

13.1 One-Sided Matching Markets

Recall that a matching market frame (n, Y, v) specifies a set $[n]$ of players, a set Y of objects (e.g., "houses") such that $|Y| = n$, and a player valuation function $v : Y \to \mathbb{N}$. Recall that from such a matching market frame, we defined a notion of a *matching market context* (n, X, T, v):

- X (the set of states) is the set of all possible allocations $a : [n] \to Y$ such that $a(i) \neq a(j)$ if $i \neq j$.
- For each player i, T_i is the set of all possible valuation functions $v : Y \to \mathbb{N}$.
- $v_i(t, a) = t(a(i))$.

As we saw in Chapter 10, for every matching market context, we can use the VCG mechanism to DST-implement social value maximization.

In this section, following the voting paradigm discussed in Chapter 11, we consider a different type of mechanism design context where players' types simply specify preference ordering over the possible objects. More formally, consider a mechanism design context (n, Ω, T, v) where:

- X is the set of all possible allocations $a : [n] \to Y$ where Y is some finite set such that $|Y| = n$, and $a(i) \neq a(j)$ if $i \neq j$.

- for each player i, the type space T_i is the set of preferences over the objects Y (i.e., each $\mu \in T_i$ is a ranking $\mu = (y_1, \ldots, y_n)$ of the elements in Y).
- for every player i, $t \in T_i$, $v_i(t, a) \geq v_i(t, a')$ if $a(i) >_\mu a'(i)$ (that is, player i is happier if it gets allocated an object it prefers).

We refer to such a context as a **matching context** (as opposed to a "matching *market* context"); if it always holds that $v_i(t, x) > v_i(t, x')$ (i.e., we have strict inequality) when x is higher ranked than x' according to t, we refer to the context as a **matching context with strict preferences**.

As with voting, we restrict our attention to payment-free mechanisms: We refer to a payment-free mechanism $\mathcal{V}$ for a matching context as a **matching rule** $\mathcal{V}$, and refer to DST matching rules as **strategy-proof matching rules**.

13.2 Strategy-proof Matching Rules: Serial Dictatorship

At first one may think that the Gibbard-Satterthwaite Theorem directly applies also to strategy-proof matching rules. Indeed, there is a sense in which it does: if players had strict preferences over the *full set of allocations*, then by the Gibbard-Satterthwaite Theorem, we would directly have that every strategy-proof matching mechanism (over at least 3 allocations) must be dictatorial. But, in a matching context, players cannot have strict preferences over the full set of allocations; rather, they can only have strict preferences over *their own* allocations (i.e., over the objects)—their utility is independent of what objects are allocated to other players!

Indeed, the type of players only specifies their own preferences over objects; as such, "dictatorship" does not even make sense as a mechanism (since even if we elect a dictator, he will only submit his own preferences, and from that we cannot deduce how to allocate objects to other players).

Indeed, as we now show, a non-trivial (and actually quite useful) mechanism is possible. Given a matching context $\Gamma = (n, \Omega, T, v)$, and given some permutation π of the players $[n]$ (i.e., an ordering of the players), consider the following mechanism "serial dictatorship mechanism" $\mathcal{V}$. Start with player $\pi(1)$ (the "first dictator") and let that player be allocated its top choice; next, move on to player $\pi(2)$ (the "second dictator"), and let that player pick its top *unallocated* choice, etc. (See Figure 13.1 for an illustration.)

We refer to this mechanisms as the **Serial Dictator Mechanism** (SDM) w.r.t. the ordering π; this mechanism is sometimes also referred to as the

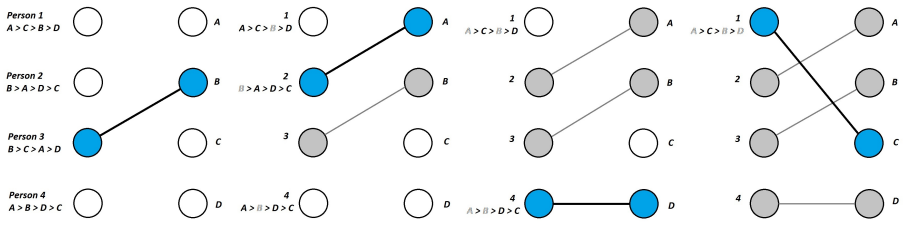


Figure 13.1: An example of the process of serial dictatorship in a matching between four players and four items. The ordering of dictators in this case is $(3, 2, 4, 1)$; each player's preferences are shown in the illustration, and at each step the current dictator selects their most preferred remaining choice (highlighted in blue).

priority mechanism.

Theorem 13.1. *For any matching context $\Gamma = (n, \Omega, T, v)$ and any ordering π of the players $[n]$, we have that SDM w.r.t. π is strategy-proof for Γ.*

Proof. Consider the "first dictator" $\pi(1)$; clearly this player gets its first choice, so can never gain anything by not truthfully reporting its preferences.

Now consider instead some later dictator $\pi(i)$ that gets to pick in the ith iteration of the mechanism. By reporting its preferences truthfully this player gets its top choice of the currently unallocated items. By deviating, it can never get an item that has already been allocated (since the mechanism never reallocates items); thus, by deviating, the player can only get allocated an item that is less (or as) desirable. ■

Not only this simple mechanism is strategy-proof, it also produced an allocation a that is "stable" in the sense that there is no allocation in which all the players are strictly better off. Assume, for contradiction, that such an allocation a' exists; clearly $a'(\pi(1)) = a(\pi(1))$ since the first dictator gets its first choice; it inductively follows that $a'(\pi(i)) = a(\pi(i))$ for every $i \in [n]$ since dictator i gets it first choice of the currently unallocated items.

Indeed, the SDM mechanism is commonly used in practice: for instance, for office allocations for professors, for the NYC school choice system, university housing allocations, etc. For all these examples, we typically select the dictatorship order π at random.

But despite all the the nice features of the SDM, there is something unappealing about it—once the order is fixed, the mechanism is very unfair. A natural question is whether this unfairness is needed.

13.3 Uniqueness of Serial Dictatorship [Advanced]

We now present a variant of the Gibbard-Satterthwaite Theorem which shows that the only strategy-proof mechanism satisfying two natural properties—*neutrality* and *non-bossiness*—is the SDM:

- We say that a matching rule $\mathcal{V}$ is **neutral** if the allocation it produces is independent of the names of the objects (that is, the outcome would be the same if the objects were renamed).

- We say that a matching rule $\mathcal{V}$ is **non-bossy** if a player cannot change the allocation of others without changing its own allocation. More precisely, for every preference profile $\vec{\mu}$, every player i, and every preference ranking μ_i' for player i, if $\mathcal{V}(\vec{\mu}) \neq \mathcal{V}(\mu_i', \vec{\mu}_{-i})$, then $\mathcal{V}(\vec{\mu})(i) \neq \mathcal{V}(\mu_i', \vec{\mu}_{-i})(i)$.

Clearly, SDM satisfies both of these properties.

We are now ready to state the theorem. As for voting contexts, we restrict our attention to matching contexts with strict preferences.

Theorem 13.2. *Let $\mathcal{V}$ be a neutral and non-bossy matching rule for a matching context with strict preferences (n, Ω, T, v). Then there exists some permutation π over $[n]$ such that $\mathcal{V}$ is the SDM w.r.t. π.*

Proof of Theorem 13.2

The proof follows a similar (but actually simpler) structure as the proof of the Gibbard-Satterthwaite theorem. Again, for simplicity of notation, in the remainder of this section, we always have some fixed matching context Γ with strict preferences in mind (and as such, whenever we write e.g., strategy-proof, neutral or non-bossy, we mean with respect to this matching market context).

We start by defining the analog of monotonicity for matching mechanisms.

Definition 13.3. A maching rule $\mathcal{V}$ is **monotone** if for every two preference profiles $\vec{\mu}$ and $\vec{\mu}'$, if $\mathcal{V}(\vec{\mu}) = a$, and for every player i and every choice y such that $a(i) >_{\mu_i} y$ it is also true that $a(i) >_{\mu_i'} y$, it follows that $\mathcal{V}(\vec{\mu}') = a$.

In other words, if a matching rule is monotone, then a winning allocation a will continue to win as long as for every player i, all candidates that are dominated by i's allocation, $a(i)$, continue to be dominated by $a(i)$.

Essentially the same proof which shows that strategy-proof voting rules are monotone shows that strategy-proof and non-bossy matching rules are monotone—the only minor difference is that we need non-bossiness to deal with the fact that player preferences are not strict over the full set of allocations (but rather only strict over each players own choices). For convenience, we repeat the proof and highlight where non-bossiness is used.

Lemma 13.4. *Any strategy-proof and non-bossy matching rule $\mathcal{V}$ is monotone.*

Proof. Assume for the sake of contradiction that $\mathcal{V}$ is not monotone; then, there exist profiles of preferences $\vec{\mu}$ and $\vec{\mu}'$ so that $\mathcal{V}(\vec{\mu}) = a$ and for every player i and every choice y such that $a(i) >_{\mu_i} y$ it is also true that $a(i) >_{\mu'_i} y$, yet $\mathcal{V}(\vec{\mu}') \neq a$.

As in the proof of Lemma 12.5, we construct a sequence of "hybrid" preferences $\vec{\mu}^k$ where we move from $\vec{\mu}$ to $\vec{\mu}'$ one player at a time:

- $\mu_i^k = \mu_i$ if $i > k$, and
- $\mu_i^k = \mu'_i$ if $i \leq k$.

Thus, $\vec{\mu}^0 = \vec{\mu}$ and $\vec{\mu}^n = \vec{\mu}'$. Now, let k^* denote the smallest index where $\mathcal{V}(\vec{\mu}^{k^*}) = a$ but $\mathcal{V}(\vec{\mu}^{k^*+1}) \neq a$. Clearly such an index must exist, since $\mathcal{V}(\vec{\mu}^n) \neq a$ by assumption.

Furthermore, note that the only difference between $\vec{\mu}^{k^*}$ and $\vec{\mu}^{k^*+1}$ is that player k^*+1 is switching from μ_{k^*+1} to μ'_{k^*+1}. *By non-bossiness, we thus must have that $a'(k^*+1) \neq a(k^*+1)$; that is, the allocation of player k^*+1 changes to some $y \neq a(k^*+1)$ when he switches strategies.*

Since $\mathcal{V}$ is strategy-proof, we must have that $a(k^*+1) >_{\mu_{k^*+1}} y$, or else player k^*+1 would prefer to report μ'_{k^*+1}, which would lead to the outcome y that they *strictly* prefer to $a(k^*+1)$ (by our assumption that preferences are strict). So, by the assumption on $\vec{\mu}'$, we have that $a(k^*+1) >_{\mu'_{k^*+1}} y$ as well.

But, by the same argument, we have that $y >_{\mu'_{k^*+1}} a(k^*+1)$ (or else player k^*+1 would prefer to report μ_{k^*+1}); this is a contradiction (since preferences are strict). ∎

We are now ready to prove Theorem 13.2.

Proof of Theorem 13.2. Consider some neutral and non-bossy matching rule $\mathcal{V}$ for n players. We show the existence of a permutation π over $[n]$ such that $\mathcal{V}$ actually implements SDM w.r.t. π.

To construct π, consider the preference profile $\vec{\nu}$ where for every player i, $\nu_i = (1, 2, 3, \ldots, n)$; that is, they all prefer $1 > 2 > 3 > \ldots > n$. Define π as follows: let $\pi(i) = j$ if $\mathcal{V}(\vec{\nu})(j) = i$; that is, let the player j who receives object i in the matching output by $\mathcal{V}($
$vec\nu$) become the "i^{th} dictator". Since every player receives a different object in the matching, this uniquely defines π.

We now show that $\mathcal{V}$ actually implements SDM_π. Consider *any* preference profile $\vec{\mu}$. We aim to show that $\mathcal{V}(\vec{\mu}) = SDM_\pi(\vec{\mu})$. Let $\vec{\nu}'$ be the preference profile where for every player i, $\nu'_i = (x_1, x_2, \ldots, x_n)$ where $x_i = SDM_\pi(\vec{\mu})(\pi(i))$ (i.e., x_i is the object received by the i^{th} dictator if running $SDM_\pi(\vec{\mu})$).

By neutrality (renaming i by x_i), it directly follows that for every i,

$$\mathcal{V}(\vec{\nu}')(\pi(i)) = x_i = SDM_\pi(\vec{\mu})(\pi(i))$$

and thus,

$$\mathcal{V}(\vec{\nu}') = SDM_\pi(\vec{\mu})$$

As we now shall argue, by monotonicity (which follows from Lemma 13.4), it also holds that

$$\mathcal{V}(\vec{\nu}') = \mathcal{V}(\vec{\mu}),$$

which will conclude the proof.

Consider the allocation $a = \mathcal{V}(\vec{\nu}')$ and for any i, the i^{th} dictator, player $\pi(i)$. As argued above, this player is currently receiving object $a(\pi(i)) = x_i$. To use monotonicity (to conclude that $\mathcal{V}(\vec{\nu}') = \mathcal{V}(\vec{\mu})$) we need to argue that every object y that is dominated by x_i w.r.t. $\vec{\nu}$ (i.e., objects x_j such that $j > i$) remains dominated by x_i w.r.t. $\vec{\mu}$.

Assume for contradiction that this is not the case; that is, there exists some x_j where $j > i$ that player $\pi(i)$ prefers to x_i w.r.t. $\vec{\mu}$. Then, the i^{th} dictator (i.e., player $\pi(i)$) in the execution $SDM_\pi(\vec{\mu})$ would be assigned x_j (since when it was their turn to choose, x_j is still unassigned), which contradicts the definition of x_i. ■

Notes

Shapley and Scarf were the first to study one-sided matching problems without payment [SS74]; the model we considered here—which typically is called the

housing allocation model—is due to Hylland and Zeckhauser [HZ79]. The serial dictatorship mechanism and the notion of "non-bossiness" were introduced and studied by Satterthwaite and Sonnenschein [SS81]. The impossibility of non-serial dictatorship mechanims is due to Svensson [Sve99]; our proof, however, is a simplified version of his proof (due to the fact that we have restricted our attention to bipartite graphs where $|X| = |Y|$).

Chapter 14

Two-sided Matchings: Stable Marriages

Let us now return to the marriage problem discussed in Section 6.5; we again consider a set X of n men, and a set Y of n women, but this time, instead of simply specifying whether a man-woman pair finds a marriage "acceptable", we consider how desirable each party finds each potential partner: for any pair $x \in X, y \in Y$, let $v_x(y)$ denote how valuable x finds y, and let $v_y(x)$ denote how valuable y finds x. We are interested in the question of whether we can find a matching between men and women where everyone is "happy" with the assignment—that is, can we find a set of "stable marriages".

14.1 Two-sided Matching Problems

To formalize this, let us first define such a matching problem, and the notion of an outcome of a matching problem.

Definition 14.1. We refer to the tuple $\Gamma = (X, Y, v)$ where X, Y are finite sets such that $|X| = |Y|$, and $v_z : X \cup Y \to \mathbb{N}$ for $z \in X \cup Y$, as a **two-sided matching problem**. An **outcome** M of Γ is a perfect matching in the bipartite graph $(X \cup Y, E)$ where E is the full set of edges (i.e., $(x, y) \in E$ for all $x \in X, y \in Y$).

Note that a two-sided matching problem can be thought of as a variant of a matching market, but this time not only the nodes on "the left" (i.e., the buyers) have a valuation for the matching, but also the nodes on "the right" (i.e., the items in the matching market) have their own valuations—this is why these types of problems are called *two-sided* matching problems.

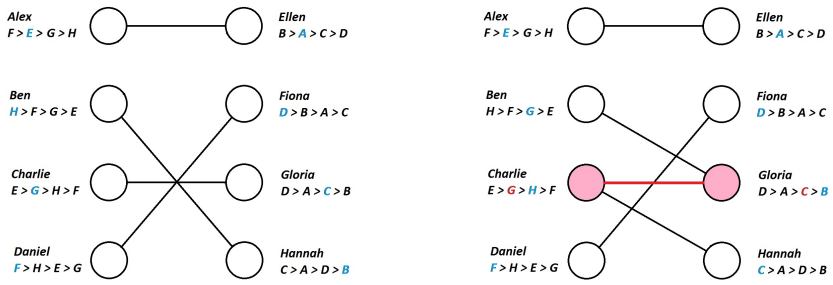


Figure 14.1: An example of a stable (left) and an unstable (right) matching between four men and four women, whose preferences are indicated in the illustration. Notice, in the unstable matching, Charlie and Gloria constitue a "blocking pair"—they both prefer each other to their current partners; this instability is highlighted in red.

We now turn to defining a notion of *stability* of an outcome of a matching problem; the notion is similar to the notion of a market equilibrium, and the notion of stability for exchange networks, but this time without any prices. Roughly speaking, an outcome (i.e., a matching M) is *stable* if there does not exists a pair (x, y) that prefer each other to their current partners—such a pair is usually called a **blocking pair**.

Definition 14.2. An outcome M of a two-sided matching problem $\Gamma = (X, Y, v)$ is **stable** if there does not exists $x \in X, y \in Y$ such that $v_x(y) > v_x(M(x))$ and $v_y(x) > v_y(M(y))$.

14.2 The Stable Marriage Theorem

The following theorem—referred as the *stable marriage theorem*—shows that every two-sided matching problem has a stable outcome. Furthermore, this stable matching can be efficiently found.

Theorem 14.3. *A stable outcome M exists for every two-sided matching problem $\Gamma = (X, Y, v)$. Additionally, there exists an efficient procedure which finds this outcome in time polynomial in $|X|$.*

Proof. We present Gale-Shapley's **deferred acceptance algorithm** (or simply, the **DA algorithm**) for finding the stable matching.

- Initialize α to the "empty" assignment: for all $x \in X, y \in Y$, $\alpha(x) = \perp$, $\alpha(y) = \perp$.
- While there exists some man x that is unmatched (i.e., $\alpha(x) = \perp$) do the following:
  - Pick some unmatched man x. Let y be x's preferred (according to v_x) "attainable" woman (breaking ties in some arbitrary way), where y is **attainable** for x if y is currently unmatched (i.e., $\alpha(y) = \perp$) or y prefers x to her current partner $\alpha(y)$ (i.e., $v_y(x) > v_y(\alpha(y))$). (Such a woman must exist since, the number of men and women are equal, and x is unmatched, so there must exists at least one unmatched woman.)
  - If y was matched, break up y's earlier "engagement": if $x' = \alpha(y) \neq \perp$, let $\alpha(x') = \perp$.
  - Match x with y; let $\alpha(x) = y, \alpha(y) = x$. (Think of this as x and y getting "engaged".)
- Output the matching M corresponding to the assignment α (note that by construction α is always one-to-one).

Note that if the algorithm ends, every man is matched to some woman (and thus also all the women get a match, since there are as many women as men).

Let us first make a simple but useful observation:

> *If a woman y is unattainable to a man x at some point in the execution of the algorithm, she will always remain so.*

This observation follows from the fact that a woman will only break an earlier engagement by "upgrading" to some man that she prefers over her current matching.

We now have the following two claims.

Claim 14.4. *The algorithm ends within $|X|^2$ iterations.*

Proof. First note that by the above observation, a man x can never be engaged to the same woman y twice—the only reason an engagement is broken off is if y "upgrades" and thus she becomes unattainable to x (and by the observation, remains so for the rest of the execution). As a consequence, each man can get engaged at most $|Y| = |X|$ times; since there are $|X|$ men, and one engagement must occur in every iteration, we conclude that the number of iterations is at most $|X|^2$. ■

Claim 14.5. *The matching M is stable.*

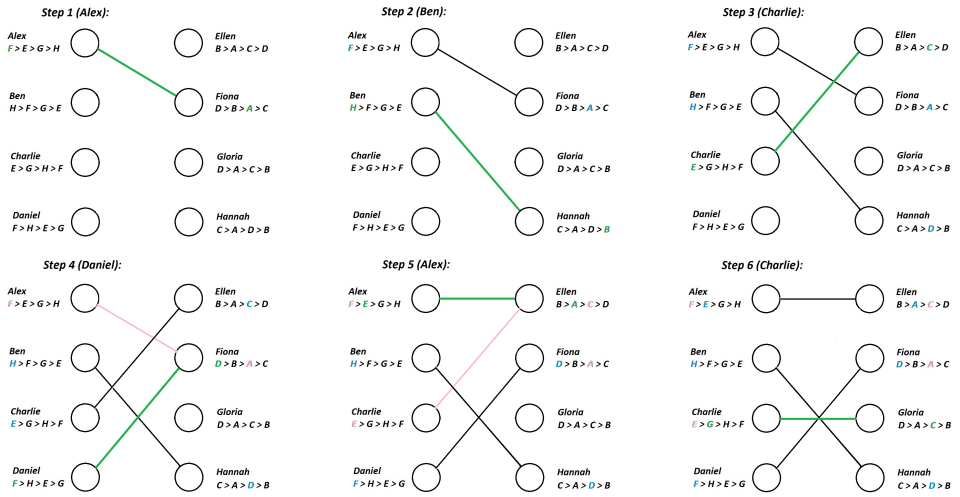


Figure 14.2: The process of running the Deferred Acceptance Algorithm on the example from Figure 14.1. New proposals are indicated in green, existing engagements in black, and broken engagements in red.

Proof. Assume, for contradiction, that there exists a blocking pair (x, y) for M; that is, there exists $x \in X, y \in Y$ such that $v_x(y) > v_x(M(x))$ and $v_y(x) > v_y(M(y))$. That means that y is attainable to x, yet x is matched with some woman $y' = M(x)$ that he likes less than y. By the above observation, since y is attainable to x at the end of the execution, she must have been so throughout the execution. By construction of the algorithm, x must thus have been engaged to y at some earlier point since he would never have proposed to y' before trying to propose to y first (and y would have accepted since she is attainable to him). But this engagement could never have gotten broken, since the only way it can break is if y no longer is attainable to x, which is a contradiction. ■

■

The above-described DA algorithm is also called the "man-proposing" DA algorithm; we may also consider an alternative variant of it, called the "woman-proposing" DA algorithm, which is identically defined except that instead of having the men propose engagements (and the women accepting or rejecting them), we instead have the women propose (and men accept or reject). Subsequently, whenever we refer to the "DA algorithm", we refer to the *man-proposing* DA algorithm.

14.3 Optimality of Stable Outcomes

We turn to analyzing the stable outcome computed by the DA algorithm. To simplify our treatment (and in accordance with much of the literature on matching), we restrict ourselves to the case when men and women have *strict* preferences over partners. Note that in this case, the DA algorithm only needs to take as input a profile of preferences $\vec{\mu}$ over partners. More precisely, given a set of men X and set of women Y, a **partner preference profile** $\vec{\mu}$ (over X, Y) as a profile of preferences such that μ_i is a preference ordering over Y if $i \in X$ and a preference ordering over X if $i \in Y$. Let $\mathcal{V}^{DA}$ denote the DA mechanism taking a partner preference profile as input.

Note that *in the case of strict preferences*, to determine whether a matching M between X and Y is stable, it suffices to know the preferences of all players $\vec{\mu}$—we can thus refer to M being a **stable matching between X, Y w.r.t. the partner preference profile $\vec{\mu}$.**

We start by showing a surprising phenomenon: for the case of strict preferences, there always exists a stable outcome that is "optimal" for the men in the sense that *every* man x gets matched to its best "achievable" partner y, where a partner is achievable for x if there exists *some* stable matching where x gets matched to y.

Definition 14.6. Given a partner preference profile $\vec{\mu}$ over X, Y, we say that $y \in Y$ is **achievable for $x \in X$ (w.r.t. $\vec{\mu}$)** if there exists some stable matching M w.r.t. $\vec{\mu}$ such that $(x, y) \in M$.

Definition 14.7. Given a partner preference profile $\vec{\mu}$ over X, Y, we say that a stable matching M is **man-optimal (w.r.t. $\vec{\mu}$)** if every man $x \in X$ gets matched (w.r.t. M) with its most preferred (w.r.t. μ_x) achievable (w.r.t. $\vec{\mu}$) woman $y \in Y$.

Note that man-optimal outcomes are always *unique* since when players have strict preferences, for every man there exists only one "most preferred" achievable woman, and this man must be matched to their most preferred acheivable choice; that is, we have the following useful fact

Fact 14.8. *Given a partner preference profile $\vec{\mu}$ over X, Y, there exists at most one man-optimal stable matching w.r.t. $\vec{\mu}$).*

We now turn to showing that the outcome produced by the (man-proposing) DA algorithm is (the unique) man-optimal matching.

Theorem 14.9. *For every partner preference profile $\vec{\mu}$, $\mathcal{V}^{DA}(\vec{\mu})$ outputs a man-optimal stable matching (w.r.t. $\vec{\mu}$).*

Proof. Assume not; let x be the first man in the execution of the DA who is "rejected" by an achievable woman y' (i.e., some achievable woman y' is no longer available when it is x's turn to get engaged). Since y' rejected x, she must already be engaged to some man x' that she prefers to x; additionally, since x was the first rejected man, y' must either be x''s optimal achievable choice, or preferred to it—in other words, from the point of view of x', y' dominates every achievable choice that is distinct from y'.

Now, consider the stable matching M where x is matched with y' (such a matching must exist, since y' is achievable for x.) Here, y' clearly prefers to switch to x', and x' also prefers to switch to y' since y' dominated every achievable choice that is distinct from y'; this contradicts stability. ■

An interesting corollary of this result is that the order in which the men get to make engagement proposals have no relevance to the outcome—we always get the *unique* man-optimal outcome.

Obviously, by the same token, the woman-proposing DA would instead give a "woman-optimal" outcome. There is, however, an inherent asymmetry (or "unfairness") in the outcomes generated by the algorithm. Man-optimal solutions, although stable, are not that great for the women, and vice versa.

Definition 14.10. Given partner preference profile $\vec{\mu}$ over X, Y, we say that a stable matching M is **woman-pessimal (w.r.t. $\vec{\mu}$)** if every woman $y \in Y$ gets matched (w.r.t. M) with its least preferred achievable man $x \in Y$.

Theorem 14.11. *Given partner preference profile $\vec{\mu}$ over X, Y, a stable matching M is man-optimal w.r.t. $\vec{\mu}$ if and only if it is woman-pessimal w.r.t. $\vec{\mu}$.*

Proof. We separately show each direction.

"only-if" direction Assume for contradiction that M is man-optimal, but not woman-pessimal; that is, there exists some woman y that is matched with some man x, but there exists some achievable man x' that y likes less than x. Consider the stable matching M' in which y is matched with the (less desirable) x' (such a matching must exists since x' is achievable for y). Let y' denote the woman that x (i.e., y's original partner) is matched with in M'. Since M was man-optimal, x must strictly prefer y to y' (by the assumption of strict preferences); and y strictly prefers x to x', thus M' cannot be stable, which is a contradiction.

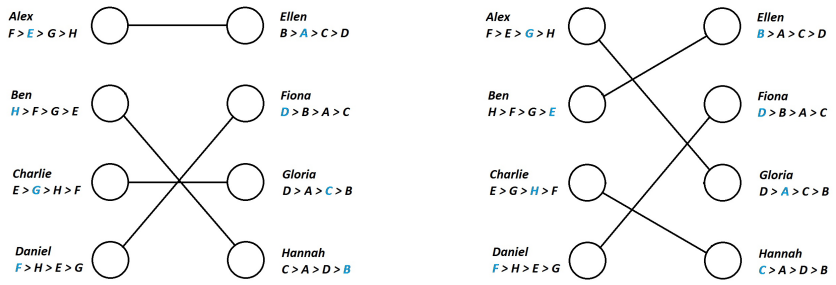


Figure 14.3: The man-optimal (left) and woman-optimal (right) stable matchings for the example from Figure 14.1.

"if" direction The backward direction follows using a similar argument, except we now show that the original matching M cannot be stable. Assume for contradiction that M is woman-pessimal, but not man-optimal; that is, there exists some man x that is matched with some woman y, but there exists some achievable woman y' that x likes more than y (i.e., $y' >_x y$). Let x' denote the man that y' is matched with in M. Since M is woman-pessimal, y' strictly prefers x to x' (by the assumption of strict preferences); however, x strictly prefers y' to y, and so M cannot be stable, which is a contradiction. ■

Thus, we see an interesting phenomenon: the order in which either the men or the women make proposals makes no difference in the final outcome, but whether the men or the women are proposers makes a big difference—the proposers get the optimal outcome, and the accepters get their pessimal outcome: it pays to be "pushy"!

14.4 Strategy-proofness of Two-sided Matching Rules

We turn to considering whether the DA algorithm is strategy-proof. Following our treatment of voting and one-sided matchings, we can define a *two-sided matching context* analogously to *matching context* except that now, a) both the left nodes and the right nodes are associated with a valuation function, and b) the players' types are preferences over partners: We say that $\Gamma = (2n, \Omega, T, v)$ is a **two-sided matching market context** if

- $n \in \mathbb{N}$ (the number of men and women)
- Ω (the set of states) is the set of matchings between $X = \{1, \ldots, n\}$ and $Y = \{n+1, \ldots, 2n\}$.
- For each player $i \in [2n]$, T_i is the set of preference orderings over Y if $i \in [n]$, and over X if $i \in [n+1, \ldots, 2n]$.
- For every player i, $t \in T_i$, $v_i(t, M) \geq v_i(t, M')$ if $M(i) >_\mu M'(i)$ (that is, player i is happier if they gets allocated a partner they prefers).

If it always holds that $v_i(t, M) > v_i(t, M')$ (i.e., we have strict inequality) when i prefers its partner in M', we refer to the context as a **two-sided matching context with strict preferences**.

We refer to a payment-free mechanism $\mathcal{V}$ for a two-sided matching-market context as a **two-sided matching rule** $\mathcal{V}$; as usual, we say that $\mathcal{V}$ is *strategy-proof* if $\mathcal{V}$ is DST. In the context of two-sided matching, we will also consider two weaker notion of strategy-proofness where DST only holds for either the men or the women:

Definition 14.12. We say that a two-sided matching rule $\mathcal{V}$ is **strategy-proof for the men** for the two-sided matching market context $\Gamma = (2n, \Omega, T, v)$ if for every $\vec{t} \in T$, $\vec{t}$ is a dominant strategy for every $i \in [n]$ in $G^{\Gamma, \vec{t}, M}$. If instead the above holds for every $i \in [n+1, \ldots, 2n]$, we say that $\mathcal{V}$ is **strategy-proof for the women**.

Clearly, a matching rule $\mathcal{V}$ is strategy-proof if and only if $\mathcal{V}$ is strategy-proof for both the men and the women. As we shall now see, the (man-proposing) DA algorithm is strategy-proof for the men in every two-sided matching context with strict preferences (and, symmetrically, the woman-proposing DA algorithm is strategy-proof for the women).

Theorem 14.13. *$\mathcal{V}^{DA}$ is strategy-proof for the men for every two-sided matching context.*

The theorem follows directly from Theorem 14.9 and the next lemma which shows that any voting rule that always outputs man-optimal outcomes is strategy-proof for the men:

Lemma 14.14. *Consider some two-sided matching context $\Gamma = (2n, \Omega, T, v)$ and a two-sided matching rule $\mathcal{V}$. Assume that for every $\vec{\mu} \in T$, $\mathcal{V}(\vec{\mu})$ is man-optimal w.r.t. $\vec{\mu}$. Then $\mathcal{V}$ is strategy-proof for the men for Γ.*

*Proof of Lemma 14.14 **[Advanced]**.* Consider a two-sided matching rule $\mathcal{V}$ that always outputs a man-optimal outcome, and assume for contradiction

that $\mathcal{V}$ is not strategy-proof for the men. That is, there exists some preference profile $\vec{\mu}$, some man i and some preferences μ_i' such that i strictly prefers its match y' in $\mathcal{V}(\mu_i', \mu_{-i})$ to its match y in $\mathcal{V}(\vec{\mu})$. We consider a set of "hybrid preferences":

- Let μ_i^0 be the preferences where y' has been "shifted up" to the top in (the "deviation") μ_i'. Note that $\mathcal{V}(\mu_i', \mu_{-i})$ (which by definition is stable w.r.t. (μ_i', μ_{-i})) must be stable also w.r.t. the "shifted" preferences (μ_i^0, μ_{-i})—i clearly does not want to change as he gets his most preferred match (i.e. y'), and thus none of the other players will want to change either (since only player i's preferences were changed). Thus, by man-optimality of $\mathcal{V}$, i's matching in $M_0 = \mathcal{V}(\mu_i^0, \mu_{-i})$ must be no worse than y', but since y' is actually i's most preferred choice, we have that $M_0(i) = y'$.

- Let μ_i^1 the preferences where y' has been shifted up to the top in the ("original") preferences μ_i. Another way to think about μ_i^1 is "morphing" μ_i^0 into μ_i while keeping the top ranked element (i.e., y') fixed; this other way will be useful to us now: Note that $\mathcal{V}(\mu_i^0, \mu_{-i})$ must be stable also w.r.t. these "morphed" preferences (μ_i^1, μ_{-i})—again: i clearly does not want to change as he gets his most preferred match (i.e. y'), and thus none of the other will want to change either (since only player i's preferences were changed). Thus, by man-optimality of $\mathcal{V}$, i's matching in $M_1 = \mathcal{V}(\mu_i^1, \mu_{-i})$ must still be y' (again, since y' is his most preferred choice).

- Note that y' must be higher ranked than y according to μ_i. Additionally, note that $M = \mathcal{V}(\vec{\mu})$ is stable w.r.t. μ_i^1, μ_{-i} since the only difference between μ_i^1 and μ_i is that we have "shifted-up" y' in μ_i^1, but this shift does not affect the set of women that i prefers to y, so i will not want to change in the shifted preferences, and consequently none of the others will either. By man-optimality of $\mathcal{V}$, it follows that no man is worse off in M_1 than in M.

- Let B be the set of men that are (strictly) better off in M_1 compared to M; all the others, $Q = X - B$ are as well off—that is, all men in Q have the same match in M and M_1. Let $\vec{\mu^2}$ denote the partner profile where for every man $x \in Q$, we have shifted up $M(x)$ to the top position in x's preferences, and let $M_2 = \mathcal{V}(\vec{\mu^2})$. First note that M is stable w.r.t. $\vec{\mu^2}$, thus by man-optimality of M_2, all men $x \in Q$ must still be matched with $M(x)$. It follows that M_2 is stable w.r.t. $\vec{\mu}$—no men in Q

want to change, and thus, none of the men in B will either (since their preferences are the same). By man-optimality of M, it thus follows that $M = M_2$.

- Consequently, we have that all the men in Q are strictly better off in M_1 than in M_2. By uniqueness of man-optimal outcomes (i.e., by Fact 14.8), we can obtain M_2 by running the DA on $\vec{\mu^2}$ in any man-proposing order. In particular, let us start by having the players in Q get their top choice, and then pick any arbitrary order. Note that the players in Q directly get their top choice, and then never change (since they need to end up with their top pick), so we can simply disregard those players and the women they get matched to.

- Additionally, note that all the remaining women (not matched to Q) must at some point in the execution of the DA algorithm reject some man (in particular, they must reject their match in M_1, since all the men in B are better off in M_1 w.r.t. $\vec{\mu}$ and thus also w.r.t. $\vec{\mu^2}$). But in the DA algorithm, the last woman who gets proposed to will never reject. Thus we have reached a contradiction.

■

14.5 Strategy-proofness v.s. Stability

So, if we use the man-proposing DA algorithm, none of the men will ever want to deviate, and if we use the woman-proposing DA algorithm, none of the women will want to deviate. We would obviously like to have a mechanism that is strategy-proof for both the men and the women. Unfortunately, as the next theorem shows, no mechanism that outputs stable outcomes can be strategy-proof (for both men and women):

Consider a two-sided matching context $\Gamma = (2n, \Omega, T, v)$ with strict preferences. We say that $\mathcal{V}$ is **stable for** Γ if $\mathcal{V}(\vec{\mu})$ is stable w.r.t. $\vec{\mu}$ for every $\vec{\mu} \in T$.

Theorem 14.15. *Consider a two-sided matching context $\Gamma = (2n, \Omega, T, v)$ with strict preferences such that $n \geq 3$. There does not exists a two-sided matching rule $\mathcal{V}$ that is both stable and strategy-proof for Γ.*

Proof. Consider some context Γ with strict preferences and assume for contradiction that there exists some two-sided matching $\mathcal{V}$ that is both stable and strategy-proof for Γ.

We start by considering the case that $n = 3$; that is, consider three men $\{M1, M2, M3\}$ and three women $\{W1, W2, W3\}$ (we may without loss of generality assume that those are their "names"), and consider the partner preference profile $\vec{\mu}$ specified as follows:

- $M1$ prefers $W1 > W2 > W3$, $M2$ prefers $W2 > W1 > W3$ and $M3$ prefers $W1 > W2 > W3$.
- $W1$ prefers $M2 > M1 > M3$, $W2$ prefers $M1 > M2 > W3$ and $W3$ prefers $M1 > M2 > M3$.

That is, $M1, M2$'s preferences are aligned—according to them the matches $(M1, W1)$, $(M2, W2)$) are optimal; likewise, $W1, W2$'s preferences are aligned but orthogonally to the men's preferences—according to them the matches $(M1, W2)$, $(M2, W1)$) are optimal. The third man and woman $(M3, W3)$ are just "dummy" players that all the others place last. Thus, there are exactly two stable matchings w.r.t. $\vec{\mu}$:

- $(M1, W1)$, $(M2, W2)$, $(M3, W3)$
- $(M1, W2)$, $(M2, W1)$, $(M3, W3)$

Assume the $\mathcal{V}(\vec{\mu})$ picks the first of these matchings (it must pick one of them since $\mathcal{V}$ is stable). If so, the men are happy with the outcome, but both the women are not. In particular, if $W1$ instead had reported $M2 > M3 > M1$, then the only stable outcome is $(M1, W2)$, $(M2, W1)$,$(M3, W3)$. This follows by simply inspecting the 6 possible matchings:

- $(M1, W1)$ can never be part of a stable matching, since then $W1$ prefers to switch to $M3$ (who prefers her).
- $(M1, W3)$ can never be part of a stable matching, since $M1$ prefers $W2$ to $W3$ (and $M1$ is $W2$'s top preference, so she would switch).
- $(M2, W3)$ can never be part of a stable matching, since $M2$ prefers $W1$ to $W3$ (and $M2$ is $W1$'s top preference, so she would switch).

Thus, we conclude that $(M1, W2), (M2, W1), (M3, W3)$ is the only stable matching under these new preferences. It follows that $\mathcal{V}$ must output it (under $W1$'s deviation), and thus $W1$ strictly gains by misreporting her preferences.

An analogous argument can be applied if the mechanism instead outputs the second matching. Finally, we remark that if $n > 3$, we can simply have man i for $i > 3$ rank woman i first, and she in turn also rank man i first.

It follows that in any stable matching, for $i > 3$, man i must be matched to woman i, and we can simply disregard these extra players and repeat the argument above. ■

We end this section by noting that Theorem 14.15 has intriguing social consequences: as we cannot hope to get a mechanism that is strategy-proof for "both sides", when running a two-sided matching mechanism, we need to decide towards what side we want robustness to manipulation. Indeed, the DA algorithm is commonly used in practice: perhaps the most famous application is for matching residents to hospitals; in this application, the residents are acting as proposers and the hospitals as accepters (which means that the mechanism is strategy-proof for the residents, but also means that the outcome is "resident-optimal").

Notes

The two-sided matching problem and the Deferred Acceptance algorithm were introduced by Gale and Shapley [GS62]; Gale and Shapley also proved the stable marriage theorem. Dubins and Freedman [DF81] and Roth [Rot82] showed the the DA algorithm is strategy-proof of the men; the proof we present here, however, is new (although is takes some inspiration from the proof in [GS85]). Roth [Rot82] showed the impossibility of mechanisms that are both strategy-proof and stable. See [RS92] for a more extensive survey of two-sided matchings. Alvin E. Roth and Lloyd S. Shapley received the Nobel prize in Economics (for the "theory of stable allocations") in 2012.

Chapter 15

Web Search

In this chapter, we consider the problem of Web search—that is, the problem of *finding the "most relevant" webpage to a given search query.* This problem is related to the sponsored search problem considered in Section 10.6, but is different in that we consider a model without payments: that is, webpages are not paying to get included in the search results. The question is thus how to design a mechanism that ensures "good search results" even if webpages try to manipulate their content to show up as a response to search queries for which they are not relevant.

For simplicity, when studying this problem, we will focus on a setting where search queries are simply keywords and assume the existence of some "naïve" scoring method for determining whether the content of a webpage is relevant with respect to that keyword—for instance, this score might depend on whether the page contains the keyword (or how many times it contains the keyword), whether it contains synonyms, spelling-corrected versions, or other related features.

Early web search algorithms relied mostly on such "content-scoring" methods to score webpages, and as a result were easy to fool by creating pages with massive lists of keywords. Modern methods for web search instead make use of the *network structure* of the Internet to "amplify" the naïve content scoring method to a more accurate score: Recall that we can view the Internet as a directed graph $G = (V, E)$—the nodes V represent webpages, and we have directed edges (v, v') between nodes (i.e., websites) v, v' if and only if v links to v'. Given such graph representation, we can specify the content of a webpages through a function $c(\cdot)$ which maps nodes in V to strings; we refer to the tuple (G, c) as a **web structure**.

15.1 Weighted Voting and PageRank

Given a naïve rule for scoring, how can we use the network structure of the Internet to improve it? The key insight is to think of the web search problem as an instance of voting.

First approach: using in-links as votes. Given some web structure (G, c), begin by removing all pages deemed irrelevant (by the content-scoring algorithm) from the graph—that is, remove all pages v whose content $c(v)$ gets a score which is below some certain threshold. Now, let a webpage's "actual score" be the number of webpages that link to it in the reduced Internet graph:

$$\text{Score}(v) = \text{in-degree}(v)$$

This already works much better than simple content-based metrics. However, it is still easy to artificially boost a page's relevance: one can simply create a lot of fake webpages (that contain a sufficient number of keywords to be deemed relevant) and link them to the webpage they wish to promote.

Weight-based voting. The obvious problem with the previous approach is that all webpages' "votes" are considered equally. This is a desideratum of elections where the voters are people, but is susceptible to the above-mentioned attack for the case of web search.

To circumvent this problem, we can assign a weight $w(v)$ to each webpage (voter) v and define the score of a webpage as follow:

$$\text{Score}(v) = \sum_{(v',v)\in E} w(v')$$

(Note that this bears some similarities to scoring-based voting rules, where the score of a candidate was defined as the sum of the weights assigned by voters to the candidates; the difference here is that voters are not assigning weights to the candidates, rather voters themselves are weighted.)

But how do we assign weights to webpages/voters so that relevant nodes are heavily weighted and spammers' "fake" webpages get a lower weight?[1] One approach—called **Hubs-and-Authorities**—introduced by Jon Kleinberg [Kle99], determines the weight of a node based on how well it is able to "predict" the relevance of a node, in the sense that the node links to the

[1] It is not absurd to ask the same question for the case of standard voting—one could potentially weigh the votes of more informed voters higher.

"relevant" webpages. More precisely, nodes are assigned two different scores: one score captures their ability as a *predictor*, and the second score captures their *relevance* as an answer to the search query; the relevance score is computed using the above-mentioned weighted scoring method, where the weight of a node is its prediction score, and the prediction score is computed in a way that gives nodes that *link* to nodes with a high relevance score, a high predicition score: the actual method is simply to let the prediction score of a node be the sum of the relevance scores of the nodes it links to. Nodes that score high as predictors are called *hubs* and nodes that score high in terms of relevance are called *authorities*. For instance, pages like Yahoo or Google News are considered hubs, whereas a content provider such as the New York Times would be considered an authority.

An alternate approach to weighting pages, introduced by Google's **PageRank** algorithm, considers a more symmetric approach: the weight of each node simply is defined to be its (relevance) score divided by the number of outgoing links. In other words, we assign more weight to links from webpages that are considered "relevant", which in turn are those that themselves are linked to "relevant" pages, and so forth; and the relevance score gets scaled by the number of outlinks a webpage has (or else, a relevant webpage could get an arbitrary number of "votes"). Formally, we have:

$$\text{Score}(v) = \sum_{(v',v)\in E} \frac{\text{Score}(v')}{\text{out-degree}(v')}$$

We can furthermore add the conditions that $\text{Score}(v) \geq 0$ for all v and that $\sum_v \text{Score}(v) = 1$.

But have we gained anything? To define the score of a node, we need to know the scores of all nodes linking to it, so isn't the definition circular? It is, but just as in the case of the definition of a Nash equilibrium, what we are looking for is a *fixed point* of the above equations—in other words, an assignment of scores to nodes such that all of the equations hold. (Recall that the notion of a Nash Equilibrium can be viewed as a strategy that itself is a fixed point of the best-response operator for all players in a game.)

Example. Consider a three-node "triangle" with nodes a, b, c and edges (a, b), (b, c), (c, a). A fixed point here would be to set all pages' scores to $1/3$—each node has one in-link and one out-link; thus,

$$\text{Score}(v) = \sum_{(v',v)\in E} \frac{\text{Score}(v')}{\text{out-degree}(v')} = \frac{1/3}{1} = 1/3$$

Figure 15.1: The example graphs consisting of a triangle with additional nodes. Notice that, if we think of PageRank score as a "fluid", it can only enter the group of purple nodes and not leave; eventually, if we iterate this process, the purple nodes' total scores will converge to 1, while the others' will approach 0.

holds for all nodes, and in addition all nodes' scores sum to 1.

An Iterative PageRank procedure Just as our method of finding PNE through BRD, we can consider an analogue of BRD in the context of PageRank. Consider the following iterative procedure:

- For each node v set $\text{Score}_0(v) = 1/n$ (where $n = |V|$).
- For each round i, set $\text{Score}_{i+1}(v) = \sum_{(v',v)\in E} \frac{\text{Score}_i(v')}{\text{out-degree}(v')}$.

So, at each step, the score of a node v is effectively split evenly among all of its out-links. We can, intuitively, consider this process in terms of a "flow" of score, where each node starts with $1/n$ unit of flow and equally distributes its flow among all outward edges.

However, this gives rise to a potential problem: If a webpage does not have any out-links, all of the "flow" it carries gets lost—there is a leak in the system! Since we must ensure that the sum of all pages' scores remains 1, we will assume that nodes without out-links at least link to themselves, and thus that all nodes in our graphs will have at least one outgoing edge. However, even with this restriction, there are some issues with this procedure.

Example. Consider the three-node graph in the prior example, but this time add a node z and edges (a, z) and (z, z) (a self-loop). (See Figure 15.1 for an illustration.) At every stage of the iterative process, z's score will increase, as it will receive "flow" from a; however, since none of z's flow will ever leave it, over time z's score will converge to 1, and the other nodes' scores

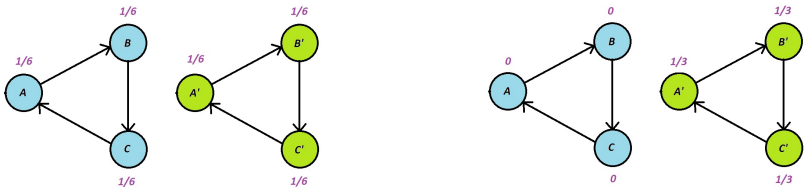


Figure 15.2: Two admissible assignments of fixed point PageRank scores for the example graph consisting of two disjoint triangles. In fact, as long as the scores of every node in a triangle are identical and the sum of all nodes' scores is 1, any such assignment is a fixed point.

to 0! In addition, this is not only a property of self-loops in the graph; we can instead add to the three-node graph two nodes, z_1, z_2, and edges, (a, z_1), (a, z_2), (z_1, z_2),(z_2, z_1). As the imaginary "flow" can never leave the network of z_1 and z_2, eventually these nodes' scores will converge to $1/2$ and the others' scores approach 0.

Example. Yet another issue with this algorithm is that the fixed points of scores in a particular graph are not uniquely defined. For instance, consider a graph consisting of two disjoint three-node triangles, a, b, c and a', b', c' (see Figure 15.2). While the iterative process gives us a fixed point where all nodes have score $1/6$, there exist infinitely many equally viable fixed points—any assignment of scores where a, b, c have score α and a', b', c' have score β, where $\alpha, \beta \geq 0$ and $3\alpha + 3\beta = 1$, is a fixed point of this network!

15.2 Scaled PageRank

Fortunately, we can refine the PageRank algorithm to deal with these problems.

ϵ-scaled PageRank. Modify the previous algorithm to include a constant parameter $\epsilon > 0$, so that, at each iteration, nodes share $(1 - \epsilon)$ of their own score with their out-links, and the remainder "evaporates" and is distributed evenly among every node. So, each node gets ϵ/n score plus whatever they receive from neighbors. (It is widely believed that Google uses $\epsilon = 1/7$ in their implementation.) In other words, we are looking for a fixed point of the

equations described by

$$\text{Score}(v) = \frac{\epsilon}{n} + (1-\epsilon) \sum_{(v',v)\in E} \frac{\text{Score}(v')}{\text{out-degree}(v')}$$

such that $\sum_v \text{Score}(v) = 1$. As before, we can attempt to find this fixed point by the following modified iterative procedure:

- For each node v set $\text{Score}_0(v) = 1/n$.
- For each round i, set $\text{Score}_{i+1}(v) = \frac{\epsilon}{n} + (1-\epsilon) \sum_{(v',v)\in E} \frac{\text{Score}_i(v')}{\text{out-degree}(v')}$.

Let us now revisit the example of the two disjoint triangles—the only fixed point is such that all nodes receive score 1/6. And, returning to the example with the node z attached to the triangle, z will no longer end up with all of the score, but instead with significantly less.

Additionally, as we now show, with this modification, scaled PageRank scores are guaranteed to exists and are uniquely defined.

Theorem 15.1. *For every $\epsilon \in (0,1)$, for every graph G where each node has at least one outgoing edge, ϵ-scaled PageRank scores are uniquely defined for each node.*

Proof. It will be convenient to adopt linear-algebraic notation to represent the network. Let us assume (without loss of generality) that the nodes of the graph are called $1, 2, \ldots, n$.

- Let $\vec{r}$ denote a vector describing the score of each node, where $r_i = \text{Score}(i)$;
- Let N be a matrix where $N_{j,i} = 0$ if there is no edge (j,i); if there is such an edge, let $N_{j,i} = \frac{1}{\text{out-degree}(j)}$

So, given this notation, for the original (non-scaled) implementation of PageRank, the fixed-point we are looking for is a score vector $\vec{r}$ such that:

$$r_i = \sum_j N_{j,i} r_j$$

or, equivalently, such that

$$\vec{r} = N^T \vec{r}$$

where N^T denotes the transpose of the matrix N. Hence, what we are looking for is an *eigenvector* of N^T with *eigenvalue 1.* (Recall that a vector $\vec{v}$ is an

eigenvector with eigenvalue λ to a matrix M if $M\vec{v} = \lambda\vec{v}$.) We must find eigenvector $\vec{r}$ that satisfies the additional constraints that 1) all entries of $\vec{r}$ are non-negative reals (since PageRank score need to be non-negative) and that the "L_1-norm" of $\vec{r}$—that is, $||\vec{r}||_1 = \sum_i |r_i|$, is equal to 1 (as the sum of all the PageRank scores need to be equal to 1). The question is whether such an eigenvector always exist?

Apowerful theorem from linear algebra, originally proposed by Oskar Perron in 1907, shows that an eigenvector $\vec{v}$ with eigenvalue 1, L_1-norm 1, and non-negative real-valued entries always exist, *and is uniquely defined*, for any matrix M satifying the following conditions:

- Every entry in M is a *strictly positive* real number.
- For any vector $\vec{x}$ with $||\vec{x}||_1 = 1$, $||M\vec{x}||_1 = 1$.

With regard to the second condition, note that multiplication by the matrix N^T corresponds to applying a single iteration of the iterative PageRank algorithm; hence, multiplying a vector whose entries sum to 1 by N^T will always produce a vector whose entries sum to 1 by the correctness of the iterative process, thus the second condition is satisfied.

However, the first condition is where the unscaled PageRank fails—recall that we set a lot of the entries of N to be zero. Now, consider scaled PageRank. We are now looking for $\vec{r}$ such that:

$$r_i = \frac{\epsilon}{n} + (1-\epsilon)\sum_j N_{j,i} r_j$$

But $||\vec{r}||_1 = 1$, so, equivalently:

$$r_i = \sum_j ((1-\epsilon)N_{j,i} + \frac{\epsilon}{n}) r_j$$

So, let us define a new matrix $\tilde{N}$ such that

$$\tilde{N}_{j,i} = (1-\epsilon)N_{j,i} + \frac{\epsilon}{n}$$

Given this new matrix, the vector $\vec{r}$ we are looking for should satisfy

$$r_i = \sum_j \tilde{N}_{j,i} r_j,$$

or, equivalently, $\vec{r} = \tilde{N}^T\vec{r}$. This time, however, by our construction of $\tilde{N}$, we know that all of the entries of $\tilde{N}^T$ are positive reals. And because multiplication by $\tilde{N}^T$ represents an iteration of the iterative scaled PageRank process,

we know it preserves the L_1-norm of a vector. So, by Perron's theorem above, we see that there is a unique fixed point $\vec{r}$ with L_1-norm 1 and non-negative entries. ■

Next, we show that the iterative procedure for scaled PageRank not only converges (to the unique scaled PageRank scores), but does so quickly.

Theorem 15.2. *The iterative procedure described above for ϵ-scaled PageRank converges the unique fixed point solution. Furthermore, at each iteration, the L_1-distance to the solution decreases by at least a factor $1 - \epsilon$.*

Proof. **[Advanced]** Let us first redefine the iterative procedure in terms of the linear-algebraic notation we used in the previous proof, as follows:

- $\vec{r^0} = \frac{1}{n}\vec{1}$
- $\vec{r^{i+1}} = \tilde{N}^T\vec{r^i} = (\tilde{N}^T)^i\vec{r^0}$

Let $\vec{r^*}$ be the uniquely defined ϵ-scaled PageRank score vector, and let $\text{Err}(t) = ||\vec{r^t} - \vec{r^*}||_1$. We show that $\text{Err}(t)$ converges to 0 as t increases, as follows:

$$\begin{aligned}
\text{Err}(t+1) = ||\vec{r^{t+1}} - \vec{r^*}||_1 = \\
||\tilde{N}^T\vec{r^t} - \tilde{N}^T\vec{r^*}||_1 = \\
||((1-\epsilon)N^T\vec{r^t} + \frac{\epsilon}{n}(\sum r_i^t)\vec{1}) - ((1-\epsilon)N^T\vec{r^*} + \frac{\epsilon}{n}(\sum r_i^*)\vec{1})||_1 \leq \\
||((1-\epsilon)N^T\vec{r^t} + \frac{\epsilon}{n}||\vec{r^t}||_1\vec{1}) - ((1-\epsilon)N^T\vec{r^*} + \frac{\epsilon}{n}||\vec{r^*}||_1\vec{1})||_1 = \\
||((1-\epsilon)N^T\vec{r^t} + \frac{\epsilon}{n}\vec{1}) - ((1-\epsilon)N^T\vec{r^*} + \frac{\epsilon}{n}\vec{1})||_1 = \\
(1-\epsilon)||N^T\vec{r^t} - N^T\vec{r^*}||_1 = \\
(1-\epsilon)||N^T(\vec{r^t} - \vec{r^*})||_1 = \\
(1-\epsilon)||(\vec{r^t} - \vec{r^*})||_1 = \\
(1-\epsilon)\text{Err}(t)
\end{aligned}$$

as desired, where the next to last equality follows from the fact that a single step of the unscaled PageRank algorithm (that is, multiplication by N^T) preserves PageRank in the system and thus the L_1 norm of any vector (and not just those with a norm of 1). (In contrast, for multiplication by $\tilde{N}^T$, this may not necessarily hold, although it does hold for vectors with norm 1.) ■

An alternate interpretation of (scaled) PageRank. Consider a process where you start off at a random node in the Internet graph, walk to a random one of its neighbors by traversing an outgoing edge, and randomly walk through the Internet in this manner (i.e. exploring the Internet at random by clicking on random links). Note that if we start of with a probability distribution over nodes, desribed as a vector $\vec{r}$ (each component of which specifies the probability of the node corresponding to the component), then the probability we obtain after one step of this process (i.e., traversing a random outgoing edge) is exactly $N^T\vec{r}$. We can now use this observation to note that the probability that you end up at a certain page after k steps is, in fact, given by k-iteration unscaled PageRank scores of the page:

- After 0 steps, we start of at the uniform distribution over nodes—that is, $\vec{r^0}$
- After 1 step, the process, we get: $N^T\vec{r}_0 = \vec{r^1}$
- After 2 steps, we get: $N^T\vec{r}_1 = \vec{r^2}$
- and so on...

So the k-iteration PageRank score of a webpage is the probability that you will end up at that page after k steps of a random walk with a random starting point. Similarly, we can think of scaled PageRank scores as characterizing a different type of random walk: At each step, with probability ϵ, you are instead transported to a random page; otherwise, you click a random outgoing link as normal.

Personalized PageRank. In the above description of scaled PageRank, we start at a uniformly random node, and each time we are randomly "transported" (with probability ϵ) we are again set back to a uniformly random node. In an alternative version, we can attempt to obtain a "personalized score" for a certain types of user by changing the distribution from being uniformly random to something else—for instance, based on how popular each webpage is (or is estimated to be) among the user's demographics.

15.3 Impossibility of Non-Manipulable Web Search

As mentioned above, pure content-scoring methods are easy to manipulate to enhance the score of certain webpages—this type of manipulation is usually referred to as **search engine optimization**. The PageRank method seems to have made the process more robust, but it is not hard to see that

also this procedure can be manipulated; indeed, there is a whole industry of companies—referred to as SEOs (search engine optimizers)—providing advice on search engine optimization. As a result, search providers (such as Google) are very secretive about the actual algorithm that they are using, and they frequently change the details of it. A natural question is whether such a "cat-and-mouse" game (between the search providers and the SEOs) is needed, or whether it is feasible to have a fully *public web search algorithm* that cannot be manipulated.

We here provide a partial answer to this question. We show that "natural" web search algorithms, of the type we have seen in this chapter, can always be manipulated. More precisely, we consider a class of **canonical web search algorithms**, which proceed as follows given a web structure $(G = (V, E), c)$ and search query Q:

- For each node v in the graph G, it applies some content-scoring method only to the content, $c(v)$, of v, to produce a content-score $s(v)$.
- Next, for each node v, compute the "final-score" (such as e.g., the PageRank score) a function only of the *graph structure*[2] of G and only the content scores $s(\cdot)$ the nodes in G (but not the actual content).

For instance, for the case of PageRank, the content-scoring method outputs either 0 or 1, based on whether the content was "relevant" to the search query, and the second step computes the PageRank of a node v by applying some function to the restricted graph obtained by removing all nodes with a content score of 0.

We now have the following result w.r.t. canonical web search algorithms: consider some web graph (G, c), some node v and some query Q such that the web search algorithm assigns v the highest score. Then, either the content-scoring method itself is already "robust" in the sense that we cannot find some "bad/spam" web content $\tilde{c}$ that would get the same content score as the real content $c(v)$, or, an SEO can "manipulate" the outcome of the search algorithm as follows: the SEO adds an independent "fake" copy of the whole web to the graph—so the new internet graph is now $G \cup G'$ where G' is an independent copy of G—except that the content of the "fake" node, v', in G' corresponding to v in G is set to $\tilde{c}$. Since a canonical web search algorithm assigns score to a node only as a function of the graph structure and content-scores of nodes, it must assign the same the score to both v and v' (since from

[2]What we mean by this is that the final score cannot depend on the names of the nodes in G; this is formalized by requiring that the final score stays the same if we permute the names of the nodes, but keep everything else (i.e., the edges and the content-score) the same w.r.t. the new names.

the perspective of the search algorothm, the nodes look identical; formally, this follows by swithing the names of all "real" nodes to their "fake" relatives, and vice versa, and noticing that this permuted graph is identical to the original, and thus v and v' must get the same scores). Thus, either of the following two things must now happen, each of which is a success for the SEO:

- the "fake" webpage v' gets a high score (i.e., shows up high in the search ranking), in which case the SEO has succeeded; or,
- the "real" v now gets a low score (i.e., no longer shows up high in the search ranking), in which case the SEO has succeeded in "taking" down a high-ranking webpage.

Thus, unless, the content-scoring method already is robust (which, as we argued before, seems hard to achieve), the web search algorithm can be manipulated by an SEO.

Of course, this attack is not easy to mount as it requires producing a copy of the whole web, so if we can impose a cost to obtaining IP-addresses, the power of it can be limited. Another approach is to rely on some content-scoring method which relies on some human intervention: for instance, the score of a webpage may become higher if the webpage has received an SSL certificate which requires manual verification of the webpage; this, again, would make it much harder for an SEO to simply copy the web. Finally, we can boostrap the problem by relying on human intervention (e.g., employees at Google) to score certain specific webpage, and simply giving lower scores to webpages that are not part of the connected components that have received manual scores—this prevents the above mentionned "cloning" attacks, as the "fake web" can now be easily distinguished from the "real web" (as the real web is connected to web pages with manual scores, whereas the fake one will not be).

Notes

As mentioned above, the "Hubs-and-Authority" approach was introduced by Kleinberg [Kle99]. Google's PageRank approach was first introduced and analyzed by Page, Brin, Motwani and Winograd [PBMW98]. The result on manipulability of web search is new.

Part IV

The Role of Beliefs

Chapter 16

The Wisdom and Foolishness of Crowds

In this chapter, we begin our explorations of the power of beliefs. We here explore how rational players update their beliefs when receiving information (signals), and how these beliefs affect the *aggregate behavior* of a crowd of people; in particular, as we shall see, depending on the context, two very different types of effects can arise. (We here make extensive use of notions from probability theory; see Appendix A for the necessary preliminaries.)

16.1 The Wisdom of Crowds

It has been experimentally observed that if we take a large crowd of people, each of whom has a poor estimate of some quantity, the *median* of the crowd's estimates tends to be a fairly good estimate: the seminal paper by Sir Francis Galton [Gal07] (who was Charles Darwin's cousin), published in *Nature* in 1907, investigated a crowd of people attempting to guess the weight of an ox at a county fair; while most people were individually far off, the "middlemost" (i.e., the median voter) was surprisingly close: the actual weight of the ox was 1198 lb and the middlemost estimate 1207 lb. And the mean of the guesses was even closer!

A Simple Signaling Game Let us introduce a simple model to explain this "wisdom of crowds" phenomenon. We will focus on a simple "yes/no" (i.e., binary) decision (e.g., "does smoking cause cancer?", or "is climate change real?"). Formally, the state of the world can be expressed as a single bit W. Now, let us assume we have a set of n individuals. *A-priori*, all of them

consider $W = 0$ and $W = 1$ to be as likely—that is, according to their beliefs $\Pr[W = 1] = 1/2$—but then each individual i receives some *independent* signal X_i that is correlated with the state of the world, but only very weakly so: for every $b \in \{0, 1\}$

$$\Pr[X_i = b \mid W = b] \geq 1/2 + \epsilon$$

where ϵ is some small constant, and all X_i are independent random variables.

Consider a game where each individual/player is supposed to output a guess for W and receives "high" utility if they output the correct answer and "low" utility otherwise.[1] The players all need to make these guesses *simultaneously* (or at least independent of the guesses of the other players), so the only information a player has when making his guess is their own signal (i.e., they cannot be influenced by the decision of others.) Intuitively, in such a situation, a player i wishing to *maximize his expected utility* should thus output X_i as his guess. To formalize this, we will rely on Bayes' Rule (see Theorem A.12 in Appedix A): Since the utility for a correct guess is higher than for an incorrect one, each player i getting a signal $X_i = 1$ should output 1 if

$$\Pr[W = 1 \mid X_i = 1] \geq \Pr[W = 0 \mid X_1 = 1]$$

Let us analyze the LHS (the left-hand side) and RHS (the right-hand side) separately. By Bayes' Rule,

$$LHS = \Pr[W = 1 \mid X_i = 1] = \Pr[X_i = 1 \mid W = 1]\frac{\Pr[W = 1]}{\Pr[X_i = 1]}$$

$$\geq \frac{(\frac{1}{2} + \epsilon)}{2\Pr[X_i = 1]}$$

By the similar logic,

$$RHS = \Pr[W = 0 \mid X_i = 1] = \Pr[X_i = 1 \mid W = 0]\frac{\Pr[W = 1]}{\Pr[X_i = 1]}$$

$$\leq \frac{(\frac{1}{2} - \epsilon)}{2\Pr[X_i = 1]}$$

which clearly is smaller than the LHS. So, we conclude that whenever a *rational player*—who wants to maximize their expected utility—receives a signal $X_i = 1$, they should output 1 as their guess; by the same argument it follows that a rational player should output 0 whenever they receive $X_i = 0$ as their signal.

[1] Note that this is not a normal-form game as we need to model the fact that players are uncertain about W and receive the signals X_i; formally, this can be done through the notion of a Bayesian game that we have alluded to in the past, but (for our purposes) this extra formalism will only add cumbersome notation without adding any new insights.

Analyzing the Aggregate Behavior What is the probability that the "majority bit" b that got the most guesses (which is equivalent to the "median vote" for the case of binary decisions) is actually equal to the true state W? At first sight, one might think that it would still only be $1/2+\epsilon$—each individual player is only weakly informed about the state of W, so why would majority voting increase the chances of producing an "informed" result? Indeed, if the signals were *dependent*, majority voting may not help (e.g., if all player receive the same signal). But, we are here considering a scenario where each player receives a "fresh", *independent*, signal. In this case, we can use a variant of the *Law of Large Numbers* to argue that, the majority vote equals W with high probability (depending on n).

To shows this, let us introduce the useful *Hoeffding bound* [Hoe63], which can be though of as a quantitative version of the Law of Large Numbers:[2]

Theorem 16.1 (Hoeffding Bound)**.** *Let $Y_1, \ldots, Y_n$ be n independent random variables over $[a, b]$. Let $M = \frac{1}{n}\sum_{i=1}^{n} Y_i$. Then:*

$$\Pr[|M - \mathbb{E}[M]| \geq \epsilon] \leq 2e^{-\frac{2\epsilon^2 n}{(b-a)^2}}$$

In other words, the Hoeffding bound is a bound on the deviation of the "empirical mean" of independent random variables (over some bounded range) from the expectation of the empirical mean. We will omit the proof of this theorem; instead, we will prove that in the above-described game, the majority vote will be correct with high probability (if players act rationally). Let $\text{Majority}(x_1, \ldots, x_n) = 0$ if at least $n/2$ of the x_i are zero, and 1 otherwise. We now have the following theorem.

Theorem 16.2. *Let $W \in \{0, 1\}$, and let $X_1, \ldots, X_n \in \{0, 1\}$ be independent random variables such that $\Pr[X_i = W] \geq 1/2 + \epsilon$. Then:*

$$\Pr[\textit{Majority}(x_1, \ldots, x_n) = W] \geq 1 - 2e^{-2\epsilon^2 n}$$

Proof. Define random variables $Y_1, \ldots, Y_n$ such that $Y_i = 1$ if $X_i = W$ and $Y_i = -1$ otherwise. Note that if

$$M = \frac{1}{n}\sum_{i=1}^{n} Y_i > 0$$

[2]For the analysis of this binary signalling game, a simpler bound called the Chernoff bound [Che52] actually suffices, but the Hoeffding bound is useful also for non-binary decisions.

then clearly Majority$(X_1, \ldots, X_n) = W$. We will show that $\Pr[M \leq 0]$ is sufficiently small, which by the above implies the theorem.
By linearity of expectations, we have

$$\mathbb{E}[M] = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}[Y_i].$$

Note that, for all i,

$$\mathbb{E}[Y_i] \geq 1(\frac{1}{2} + \epsilon) + (-1)(\frac{1}{2} - \epsilon) = \frac{1}{2} + \epsilon - \frac{1}{2} + \epsilon = 2\epsilon.$$

Thus,

$$\Pr[M \leq 0] \leq \Pr[|M - 2\epsilon| \geq 2\epsilon] = \Pr[|M - \mathbb{E}[M]| \geq 2\epsilon]$$

which by the Hoeffding bound (setting $a = 1$, $b = -1$) is smaller than

$$2e^{-\frac{2(2\epsilon)^2 n}{(b-a)^2}} = 2e^{-\frac{2*4\epsilon^2 n}{2^2}} = 2e^{-2\epsilon^2 n}$$

■

Observe that, as we might expect, the (lower bound on the) probability that the majority of the guesses is correct increases with a) the number of players n, and b) the probability ϵ that individual players' guesses are correct.

A connection to two-candidate voting. The above theorem shows that the majority voting rule (studied in Chapter 11) has the property that, if the outcome over which players vote objectively is "good" or "bad" (but players are uncertain about which is the case), majority voting will lead to the "good" outcome as long as players receive independent signals sufficiently correlated with the correct state of the world.

Beyond binary decisions Let us briefly point out that the Hoeffding bound applies also in a setting where players need to guess some (non-binary) real number in some *bounded interval* (like in the "guessing the weight of the ox" example). Consider, for instance, a scenario where everyone receives as a signal an independently "perturbed" version of the "true value", where the expectation of the perturbation is 0 (i.e., positive and negative perturbations are as likely)—that is, everyone gets an independently drawn signal whose expected value is the "true value". The Hoeffding bound then says that, with high probability, the mean of the signals of all the players is close to the true value.

16.2 The Foolishness of Crowds: Herding

Let us now return to the binary decision scenario, but let us change the problem a bit: Instead of having the players *simultaneously* announce their guesses, the players announce *sequentially*, one after the other; additionally, before making their own guess, each player gets to first observe the guesses of everyone who preceeded them. (For instance, think of this as posting your opinion on Facebook after first seeing what some of your friends posted.)

How should people act in order to maximize the probability that their own guess is correct (and thus maximize their expected utility)? (Again, formally, we can model the situation as a game where the player receives some high utility being for a correct guess and a low utility for an incorrect guess, but this time it is a game over multiple stages (i.e., an *extensive-form game*).) As before, *a-priori*, the players consider $W = 0$ and $W = 1$ to be as likely; that is, according to their beliefs $\Pr[W = 1] = 1/2$. Let us now additionally assume that all players' evidence is *equally strong*—that is, for $b \in \{0, 1\}$,

$$\Pr[X_i = b \mid W = b] = 1/2 + \epsilon$$

for all i (i.e., we have equality instead of $\geq$). Most importantly, let us now also assume that players not only are rational themselves, but that it is *commonly known* that everyone is rational—that is, everyone is rational, everyone knows that everyone is rational, everyone knows that everyone knows that everyone is rational, etc. We shall see how to formalize this notion of common knowledge of rationality in Chapter 17, but for now, we appeal to intuition (which will suffice for this example).

Analyzing the Aggregate Behavior Let us analyze the players one-by-one:

- For the first player, nothing has changed from before: the only information he has is X_1 and thus (by the same arguments as above) he will maximize his utility by guessing $g_1 = X_1$.

- The second player effectively has two pieces of evidence now—their own and also the guess of the first player. Additionally, since the second player *knows that the first player is rational*, he knows that $g_1 = X_1$. Thus, the second player now effectively sees the signals X_1, X_2. Intuitively, if $X_1 = 1$ and $X_2 = 1$, they should guess $g_2 = 1$, since they now have two pieces of evidence favoring $W = 1$. We again rely on Bayes'

rule to formalize this. Since the second player is rational, they should guess $g_2 = 1$ after seeing $X_1 = 1, X_2 = 1$ if:

$$\Pr[W = 1 \mid X_1 = X_2 = 1] \geq \Pr[W = 0 \mid X_1 = X_2 = 1]$$

By Bayes' Rule,

$$\Pr[W = 1 \mid X_1 = 1, X_2 = 1] = \Pr[X_1 = 1, X_2 = 1 \mid W = 1] \frac{\Pr[W = 1]}{\Pr[X_1 = 1, X_2 = 1]}$$

which by independence of the signals X_1, X_2 equals

$$\frac{(\frac{1}{2} + \epsilon)^2}{2 \Pr[X_1 = 1, X_2 = 1]}$$

By the same logic,

$$\Pr[W = 0 \mid X_1 = 1, X_2 = 1] = \frac{(\frac{1}{2} - \epsilon)^2}{2\Pr[X_1 = 1, X_2 = 1]}$$

which clearly is smaller. So, if the second player sees $X_1 = 1, X_2 = 1$ they should output 1, and if they see $X_1 = 0, X_2 = 0$ they should output 0. If $X_1 = 1$ but $X_2 = 0$, or vice versa, then intuitively $W = 0$ and $W = 1$ are equally likely (which again can be formalized using Bayes' Rule), so a rational player can output either choice as its guess. We will assume that a rational player will prefer his own signal and output $g_2 = X_2$ in this case. Thus, we conclude that a rational player 2, *who knows that player 1 is rational* (and thus $g_1 = X_1$), will always output his signal as his guess (just as a rational player 1 would).[3] So far, all is good...

- Let us now turn to the third player. Consider the scenario where the third player sees $g_1 = 1, g_2 = 1$. In this case, *no matter what X_3 is, the third player will be better off guessing 1* under our rationality assumptions. Specifically, if they see three independent signals of equal strength, two of which point to either $W = 0$ or $W = 1$, it is always better in expectation for the third player to ignore their own evidence and guess in line with the majority. This follows using exactly the same analysis with Bayes' rule as before. For instance, to analyze the case when the third player's own signal $X_3 = 0$, by the previous argument:

$$\Pr[W = 1 \mid X_1 = 1, X_2 = 1, X_3 = 0] = \frac{(\frac{1}{2} + \epsilon)^2(\frac{1}{2} - \epsilon)}{2\Pr[X_1 = 1, X_2 = 1, X_3 = 0]}$$

[3] In fact, at this point (although it makes the argument cleaner) we do not even have to asume that player 2 knows that player 1 is rational, since no matter what X_1 is, player 2 can rationally output $g_2 = X_2$.

and

$$\Pr[W = 0 \mid X_1 = 1, X_2 = 1, X_3 = 0] = \frac{(\frac{1}{2} - \epsilon)^2(\frac{1}{2} + \epsilon)}{2\Pr[X_1 = 1, X_2 = 1, X_3 = 0]}$$

which is smaller. Thus, we see that if the third player believes that the earlier two players output their signals as their guesses, which follows the assumption that player 3 knows that a) player 1 is rational (and thus $g_1 = X_1$), and b) player 2 is rational and knows that player 1 is rational (and thus $g_2 = X_2$) that in the event that they see $g_1 = 1, g_2 = 1$, they should *ignore their own signal* and simply guess 1 (and analogously if they see $g_1 = 0, g_2 = 0$).

- Of course, should $g_1 = 1, g_2 = 1$ happens, then player 4, knowing that player 3 ignored their own evidence (by our rationality assumption), would be in exactly the same situation as player 3 and should thus *also ignore their own evidence* and output $g_4 = 1$ as well. And so, we end up with an **information cascade**, where everyone guesses in accordance with the first two players and ignores their own evidence. (In general, this will occur at any point where the number of guesses for either 0 or 1 outnumbers the other by 2!)

Notice that, if, say, $W = 0$, a cascade where *everyone guesses the incorrect state* happens as long as $X_1 = 1, X_2 = 1$, which occurs with probability $(\frac{1}{2} - \epsilon)^2$. Even with a relatively large ϵ—say, $\epsilon = 0.1$—this would still occur with probability $0.4^2 = 0.16$. So, to conclude, if rational players make their guesses *sequentially*, rather than in parallel, then with probability $(\frac{1}{2} - \epsilon)^2$, not only will the majority be incorrect, but we get a **herding behavior** where *everyone* "copies" the first two player and thus guesses incorrectly!

In fact, the situation is even worse: if $X_1 \neq X_2$, the remaining players are effectively ignoring the outputs of the first two players and "start over". Thus, we will eventually (in fact, rather fast), get to a situation where a cascade starts, and with probability close to $1/2$ the cascade is "bad" in the sense that all the cascading players output an incorrect guess.[4].

Of course, in real life, decisions are never fully sequentialized. But a similar analysis applies as long as they are sufficiently sequential—in contrast, if they are sufficiently parallelized, then the previous section's "wisdom of crowds" analysis applies.

[4]Formally, the probability of the cascade being bad is $\frac{(1/2-\epsilon)^2}{(1/2-\epsilon)^2+(1/2-\epsilon)^2}$

Let us end this section by pointing out that there are several other reasons why cascades may not happen as easily in real life (even if decisions are sequentialized). For instance, in real life, people have a tendency to weight their own evidence (or, for instance, that of their friends or relatives) more strongly than others' opinions or evidence, whereas in the above-described model, rational agents weight every piece of evidence equally. Furthermore, the herding model disregards the ability of agents to acquire new evidence—for instance, if the third player knows that their own evidence will be ignored no matter what, then they may wish to procure additional evidence at a low cost, if the option is available.

Notes

The term "the wisdom of crowds" was coined in [Sur05] by Surowiecki; Surowiecki's book also contains many case studies which illustrate the phenomena that the aggregation of information in groups results in decisions that often are better than could have been made by any single member of the group.

Herding was first explored by Banerjee's in [Ban92]; Banerjee's analysis however relied on stronger rationality assumptions than we do here.

Chapter 17

Knowledge and Common Knowledge

As we have already seen in Chapter 16, reasoning about players knowledge about *other players' knowledge* was instrumental in understanding herding behavior. In this chapter, we introduce formal models for reasoning about such *higher-level knowledge (and beliefs)*. But before doings so, let us first develop some more intuitions, through the famous "muddy children" puzzle.

17.1 The Muddy Children Puzzle

Suppose a group of children are in a room. Some of the children have mud on their foreheads; all of the children can see any other child's forehead, but they are unable to see, feel, or otherwise detect, whether they themselves have mud on their own forehead. Their father enters the room, announces that some of the children have mud on their foreheads, and asks if anyone *knows for sure* that they have mud on their forehead. All of the children say "no".

The father asks the same question repeatedly, but the children continue to say "no", until, suddenly, on the tenth round of questioning, *all of the children with mud on their foreheads answer "yes"!* How many children had mud on their foreheads?

The answer to this question is, in fact, ten. More generally, we can show the following (informally stated) claim:

Claim 17.1 (informal)**.** *All of the muddy children will say "yes" in, and not before, the n^{th} round of questioning if and only if there are exactly n muddy children.*

Proof. (informal) We here provide an informal inductive argument appealing to "intuitions" about what it means to "know" something—later, we shall formalize a model of knowledge that will enable a formal proof. Let $P(n)$ be the statement that the claim is true for n children. That is, we're trying to prove that $P(n)$ is true for all $n \geq 1$. We prove the claim by induction.

Base case: We begin by showing $P(1)$. Because the father mentions that there are some muddy children, if there is only one muddy child, they will see nobody else in the room with mud on their forehead and know in the first round that they are muddy. Conversely, if there are two or more muddy children, they are unable to discern immediately whether they have mud on their own forehead; all they know for now is that some children (which may or may not include themselves) are muddy.

The Inductive step: Now assume that $P(k)$ is true for $0 \leq k \leq n$; we will show $P(n+1)$. Suppose there are exactly $n+1$ muddy children. Since there are more than n muddy children, by the induction hypothesis nobody will say "yes" before round $n+1$. In that round, each muddy child sees n other muddy children, and knows thus that there are either n or $n+1$ muddy children total. However, by the induction hypothesis, they are able to infer that, were there only n muddy children, someone would have said "yes" in the previous round; since nobody has spoken yet, each muddy child is able to deduce that there are in fact $n+1$ muddy children, including themselves.

If there are strictly more than $n+1$ muddy children, however, then all children can tell that there are at least $n+1$ muddy children just by looking at the others; hence, by the induction hypothesis, they can infer from the start that nobody will say "yes" in round n. So they will have no more information than they did initially in round $n+1$, and will be unable to tell whether they are muddy as a result. ■

17.2 Kripke's "Possible Worlds" Model

Let us now introduce a model of knowledge that allows us to formally reason about these types of problems. We use an approach first introduced by the philosopher Saul Kripke. To reason about beliefs and knowlede, the idea is to consider not only facts about the "actual" world we are in, but also to consider a set of "possible" worlds Ω. Each possible world, $\omega \in \Omega$, specifies some "outcome" $s(\omega)$; think of the outcome as the set of "ground facts" that we care about—for instance, in the muddy children example, $s(\omega)$ specifies which

children are muddy. Additionally, each world specifies "beliefs" for all of the players; for each player i, player i's beliefs at world ω—denoted $P_i(\omega)$—are specified as a *set of worlds*; think of these as the worlds that player i *considers possible at* ω (i.e., worlds that i cannot rule out as being impossible according to him). We now, intuitively, say that a player i *knows* some statement ϕ at some world ω if ϕ is true in *every world* i *considers possible* at ω (i.e., ϕ holds at all the world in $P_i(\omega)$.

Knowledge Structures Let us turn to formalizing the possible worlds model, and this notion of knowledge.

Definition 17.2. A (finite) knowledge structure is a tuple $M = (n, \Omega, X, s, \vec{P})$ such that:

- $[n]$ is a finite set of players.
- Ω is a finite set; we refer to Ω as the set of "possible worlds" or "possible states of the world".
- X is a finite set of outcomes.
- $s : \Omega \to X$ is a function that maps worlds $\omega \in \Omega$ to outcomes $x \in X$.
- For all $i \in [n]$, $P_i : \Omega \to 2^\Omega$ maps worlds to sets of possible worlds; we refer to $P_i(\omega)$ as the beliefs of i at ω—that is, these are the worlds that i considers possible at ω.
- For all $\omega \in \Omega$, $i \in [n]$, it holds that $\omega \in P_i(\omega)$. (That is, in every world ω, players always consider that world possible.)
- For all $\omega \in \Omega$, $i \in [n]$, and $\omega' \in P_i(\omega)$, it holds that $P_i(\omega') = P_i(\omega)$. (That is, at every world ω, in every world players consider possible, they have the same beliefs as in ω.)

Definition 17.3. Given a knowledge structure $M = (n, \Omega, X, s, \vec{P})$, we define an **event**, ϕ, in M as a subset of states $\omega \in \Omega$ (think of these as the set of state where "ϕ holds"). We say that ϕ **holds at a world** ω if $\omega \in \phi$. Given an event ϕ, define the event $\mathbf{K}_i^M \phi$ ("player i knows ϕ") as the set of worlds $\omega \subseteq \Omega$ such that $P_i(\omega) \subseteq \phi$. Whenever M is clear from the context, we simply write $\mathbf{K}_i \phi$. Finally, define the event $\mathbf{E}\phi$ ("everyone knows ϕ") as $\bigcap_i \mathbf{K}_i \phi$.

The last two conditions in Definition 17.2 deserve some clarification. Given our definition of knowledge, the last condition in Definition 17.2 is equivalent to saying that players "know their beliefs"—it requires that at every world ω, in every world they consider possible (in ω), their beliefs are the same as in ω (and thus at ω, players "know their beliefs at ω"). That players know their beliefs is an important consistency condition; if it did not hold, some player

must consider it possible that they consider something possible which they actually do not consider possible!

The second-to-last condition, is easier to interpret. It says that at any world ω, players never rule out (the true world) ω. Intuitively, if ω is the true state of the world, players can never has seen evidence that ω is impossible and thus they should never rule it out. In particular, this condition means that player can never know something that is not true: if something holds in every world a player considers possible, then it must hold in the actual world, since the actual world is deemed possible. While this condition is appropriate for defining *knowledge*, it may not always be the best way to model *beliefs*—indeed, sometime people may sometimes be fully convinced of false statements! We will return to this point later when discussing the difference between knowledge and beliefs.

Knowledge Partitions Let us point out a useful consequence of the two conditions we just discussed. A direct consequence of last condition is that for each player i, and any two worlds ω_1, ω_2, we have that either $P_i(\omega_1) = P_i(\omega_2)$ or $P_i(\omega_1)$ and $P_i(\omega_2)$ are disjoint. The second-to-last condition, in turn, implies that the beliefs of a player cover the full set of possible world. Thus, taken together, we have that the beliefs of a player *partitions* the states of the world into different disjoint "cells" where the beliefs of the player are all the same—these cells are sometime referred to as the **knowledge partitions** or **knowledge cells** of a player.

Knowledge Networks To reason about knowledge structures it is convenient to represent them as "knowledge networks/graphs": The nodes of the graph corresponds to the possible state of the world, $\omega \in \Omega$; we label each node ω by the outcome, $s(\omega)$, in it. The edges in the graph represent the players' beliefs: we draw a directed edge with label i between nodes ω, ω' if $\omega' \in P_i(\omega)$ (that is, if player i considers ω' possible in ω). Note that by the second-to-last condition in Definition 17.2, each nodes has a self-loop labelled by every player identity $i \in [n]$.

To gain some familiarity with knowledge structures and their network representations, let us consider two examples. We first consider a simple example in the context of single-good auctions, and then return to the muddy children puzzle.

Example (A Single-good Auction). Consider a single-item auction with two buyers. Let the outcome space $X = \mathbb{N} \times \mathbb{N}$—each outcome $(v_1, v_2) \in X$

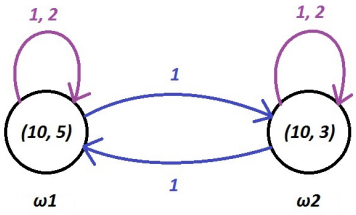


Figure 17.1: A knowledge network for a single-item auction, where the world state is dependent on player 2's valuation. This captures the fact that player 2 is fully aware of his valuation, but player 1 is not.

specifies the valuation each player has for the item. Consider a knowledge structure with just two possible worlds, ω_1 and ω_2, such that $s(\omega_1) = (10, 5)$, $s(\omega_2) = (10, 3)$, and $P_1(\omega_1) = P_1(\omega_2) = \{\omega_1, \omega_2\}$, $P_2(\omega_1) = \omega_1$, and $P_2(\omega_2) = \omega_2$. See Figure 17.1 for the knowledge network corresponding to this knowledge structure.

Note that in both worlds, player 1 knows his own valuation (10), but he does not know player 2's valuation—as far as he knows it may be either 3 or 5. In contrast, player 2 knows both his own and player 1's valuation for the item in both worlds.

Also, note that if we were to change the structure so that $P_1(\omega_2) = \{\omega_2\}$, this would no longer be a valid knowledge structure, as the last condition in Definition 17.2 is no longer satisfied (since $\omega_2 \in P_1(\omega_1)$ but $P_1(\omega_1) \neq P_1(\omega_2)$); in particular, in ω_1 player 1 would now consider it possible that he knows player 2's valuation is 3 whereas he does not actually know it.

Note that a world determines not only players' beliefs over outcomes (valuations, in the above example), but also players' beliefs about *what other players believe* about outcomes. We refer to these beliefs about beliefs (or beliefs about beliefs about beliefs..., etc.) as players' **higher-level beliefs**. Such higher-level beliefs are needed to reason about the muddy children puzzle, as she shall now see.

Example (the Muddy Children) Assume for simplicity that we have two children. To model the situation, consider an outcome space $X = \{M, C\}^2$ (i.e., an outcome specifies whether each of the children is muddy (M) or clean (C)), and the set of possible worlds $\Omega = \{\omega_1, \ldots, \omega_4\}$ such that:

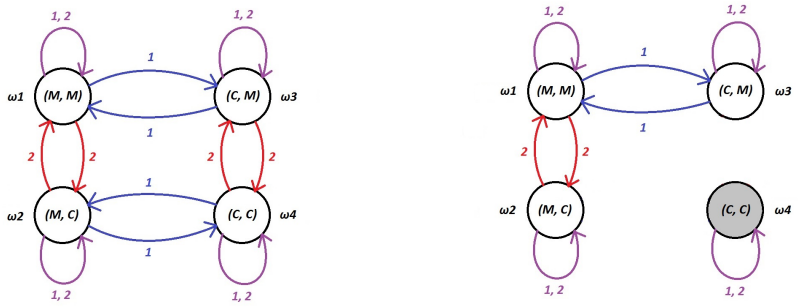


Figure 17.2: Left: the knowledge network for the muddy children example. Right: The knowledge network when the father announces that at least one child is muddy. Notice that, unless the state is ω_1, one of the two children is now able to determine the current state.

- $s(\omega_1) = (M, M)$
- $s(\omega_2) = (M, C)$
- $s(\omega_3) = (C, M)$
- $s(\omega_4) = (C, C)$

By the rules of the game, we have some restrictions on each player's beliefs. Player 1's beliefs are defined as follows:

- $P_1(\omega_1) = P_1(\omega_3) = \{\omega_1, \omega_3\}$ (player 1 knows player 2 is muddy, but cannot tell whether or not he himself is);
- $P_1(\omega_2) = P_1(\omega_4) = \{\omega_2, \omega_4\}$ (player 1 knows player 2 is clean, but cannot tell whether or not he himself is).

Analogously for player 2:

- $P_2(\omega_1) = P_2(\omega_2) = \{\omega_1, \omega_2\}$;
- $P_2(\omega_3) = P_2(\omega_4) = \{\omega_3, \omega_4\}$.

The knowledge network corresponding to this sitution is depicted in Figure 17.2. To reason about this game, define the event MUDDY_i as the set of worlds where player i is muddy, and let the event $\mathsf{SOMEONE\_MUDDY} = \bigcup_i \mathsf{MUDDY}_i$. Consider now a situation where both children are muddy; that is, the true state of the world, ω is ω_1. Notice that

$$\omega \in \mathbf{K}_1 \, \mathsf{SOMEONE\_MUDDY}$$

since player 1 knows that someone (player 2) is muddy (recall that $\mathbf{K}_1\mathsf{SOMEONE\_MUDDY}$ denotes the set of states where player 1 knows that the event $\mathsf{SOMEONE\_MUDDY}$ holds). Similarly,

$$\omega \in \mathbf{K}_2\, \mathsf{SOMEONE\_MUDDY}$$

and thus, we have

$$\omega \in \mathbf{E}\ \mathsf{SOMEONE\_MUDDY}$$

Furthermore, note that for both players $i \in [2]$,

$$\omega \in \neg\mathbf{K}_i\, \mathsf{MUDDY}_i$$

where for any event ϕ, $\neg\phi$ denotes the complement of the event ϕ, since nobody initially knows whether they are muddy or not.

Common Knowledge Does the father's announcement that there is some muddy child tell the children something that they did not already know? At first sight it may seem like it does not: the children already knew that some child is muddy! However, since the announcement was *public*, they do actually learn something new, as we shall now see.

Before the announcement, player 1 considers it possible that player 2 considers it possible that the true state of the world is ω_4 and thus that nobody is muddy. But after the announcement, this is no longer possible: we know that under no circumstances, ω_4 can never be deemed possible; that is, in the knowledge network, we should now remove all arrows that point to ω_4, as seen in the right of Figure 17.2. In particular, we now have that everyone knows that someone is muddy (which previously was not known); that is,

$$\omega \in \mathbf{E}\ \mathbf{E}\ \mathsf{SOMEONE\_MUDDY}$$

In fact, since the announcement was public, the fact that "someone is muddy" becomes *commonly known*—that is, everyone knows it, everyone knows that everyone knows it, everyone knows that everyone knows that everyone knows it, etc. We formalize this notion in the straight-forward way:

Definition 17.4. Let $\mathbf{C}\ \phi$ ("ϕ is common knowledge") be the event $\bigcap_i \mathbf{E}^i\, \phi$.

So, after the announcement, we have

$$\omega \in \mathbf{C}\ \mathsf{SOMEONE\_MUDDY}$$

The important takeaway here is that:

- if a statement ϕ gets *communicated in private*, the statement becomes known to the receiver; whereas,
- if the statement is *publicly announced*, the statement gets commonly known among the recipients.

There in an intresting way to characterize common knowledge using the knowledge network corresponding to the knowledge structure: common knowledge of ϕ holds at a state ω (i.e., $\omega \in \mathbf{C}\phi$) if and only if ϕ holds at every state that is reachable from ω in the knowledge network. (We leave it as an exercise to prove it.)

Back to the Muddy Children: Returning to the muddy children puzzle, does the announcement of the father enable anyone to know whether they are muddy? No, in fact; it follows by inspection that even in the graph without ω_4, we still have for $i \in [2]$,

$$\omega \in \neg \mathbf{K}_i \, \mathsf{MUDDY}_i$$

So, everyone will reply "no" to the first question. How do these announcements (of the answers "no") change the players' knowledge? The players will now know that states ω_2 and ω_3 are impossible, since $\omega_2 \in \mathbf{K}_1 \, \mathsf{MUDDY}_1$ and $\omega_3 \in \mathbf{K}_2 \, \mathsf{MUDDY}_2$; hence, if either of these states were true, one of the two children would have answered "yes" to the first question.

So, when the father asks the second time, the graph is reduced to only the state ω_1, and so both children know that they are muddy and can answer "yes". The same analysis can be applied to a larger number of players; by using an inductive argument similar to the one at the start of the chapter, we can prove using this formalism that after k questions all states where k or fewer children are muddy have been removed. And so, in any state where there exist m muddy children, all of them will respond "yes" to the m^{th} question.

17.3 Can We Agree to Disagree? [Advanced]

Consider some knowledge structure $\Gamma = (n, \Omega, X, s, \vec{P})$, and let f be some probability mass function over the set of states Ω; think of f as modeling players' "prior" beliefs "at birth" before they receive any form of information/signals about the state of the world. This probability mass function is referred to as the **common prior** over states, as it is common to all players.

In a given world $\omega \in \Omega$, a player i may have received additional information and then potentially "knows" that not every state in Ω is possible—it now

only considers states in $P_i(\omega)$ possible. Consequently, we can now define a player i's (subjective) **posterior beliefs** in ω by conditioning f on the set of states $P_i(\omega)$. Concretely, given some event H,

- Let $[\Pr(H) = \nu]_i$ (read "i assigns probability ν to the event H") denote the set of states ω such that $\Pr_{(\Omega,f)}[H \mid P_i(\omega)] = \nu$); that is the set of states where
$$\frac{\sum_{\omega' \in H \cap P_i(\omega)} f(\omega')}{\sum_{\omega' \in P_i(\omega)} f(\omega')} = \nu$$
holds.

A natural question that arises is whether it can be *common knowledge that two players disagree on the probability of some event holding.* Aumann's "agreement theorem" shows that this cannot happen. Formally,

Theorem 17.5. *Let $M = (n, \Omega, X, s, \vec{P})$ be a knowledge structure, and let f be a common prior over Ω. Suppose there exists some world $\omega \in \Omega$ such that*

$$\omega \in \mathbf{C}([\Pr(H) = \nu_i]_i \cap [\Pr(H) = \nu_j]_j).$$

Then, $\nu_i = \nu_j$.

Before proving this theorem, let us first state a simple lemma about probabilities:

Lemma 17.6. *Let $F_1, \ldots F_m$ be disjoint events and let $F = F_1 \cup F_2 \ldots \cup F_m$ (that is, $F_1, \ldots, F_m$ partitition, or "tile", the set F). Then, for eny event H, if for all $i \in [m]$ we have that $\Pr[H \mid F_i] = \nu$, it follows that $\Pr[H \mid F] = \nu$.*

Proof. The lemma follows directly by expanding out the definition of conditional probabilites. In particular, by Claim A.11, we have:

$$\Pr[H \cap F] = \sum_{i \in [m]} \Pr[H \cap F \mid F_i] \Pr[F_i] = \sum_{i \in [m]} \Pr[H \mid F_i] \Pr[F_i] = \\ = \sum_{i \in [m]} \nu \Pr[F_i] = \nu \Pr[F]$$

which concludes the claim. ■

We now turn to the proof of the agreement theorem.

Proof of Theorem 17.5. Let $H_{i,j}^{\nu_i,\nu_j} = [\Pr(H) = \nu_i]_i \cap [\Pr(H) = \nu_j]_j$ denote the event that i assigns probability ν_i to H and j assigns probability ν_j to H. Consider some state ω where $H_{i,j}^{\nu_i,\nu_j}$ is common knowledge; that is,

$$\omega \in \mathbf{C}H_{i,j}^{\nu_i,\nu_j}$$

As noted above, this means that $H_{i,j}^{\nu_i,\nu_j}$ holds at every state that is "reachable" from ω in the network graph; let C denote the set of states that are reachable from ω.

Let $\omega_1^i, \ldots \omega_{m_i}^i$ denotes a set of states such that the beliefs of i in those states "tile" all of C; that is,

$$C = P_i(\omega_1^i) \cup P_i(\omega_2^i) \ldots \cup P_i(\omega_{m_i}^i)$$

and for any two k, k', we have that $P_i(\omega_{k_1}^i)$ is disjoint from $P_i(\omega_{k_2}^i)$. Since, as noted above, the beliefs of a player *partition* the states of the world, such a set of states is guaranteed to exist. Since $H_{i,j}^{\nu_i,\nu_j}$ holds at every state in C, we have that for every $k \in [m_i]$,

$$\Pr[H \mid P_i(\omega_k^i)] = \nu_i$$

By Lemma 17.6 (since the beliefs "tile" C), it follows that $\Pr[H \mid C] = \nu_i$. But, by the same argument, it also follows that $\Pr[H \mid C] = \nu_j$ and thus we have that $\nu_i = \nu_j$. ■

Let us remark that it suffices to consider a knowledge structure with just two players—thus, when we say that it is common knowledge that the players disagree on the probability they assign to some event, it suffices to say that it is common knowledge *only among them.* In other words, the players cannot "agree that they disagree" on the probability they assign to the event.

17.4 The "No-Trade" Theorem [Advanced]

So far, when we have been discussing markets and auctions, we have not considered the value the seller has for the item he/she is selling, and thus whether the seller is willing to go through with the sale. This was actually without loss of generality, since the seller could simply also act as an additional buyer to "buy-back" the item in case nobody is bidding high enough.

However, in our treatment so far we assume that 1) the players *always know* how much the item is worth to them, and 2) the players may have different *private valuations* of items. Indeed, the reason the trade takes place

is because of the difference in these private valuations. For instance, if the seller of a house in NYC is moving to California, the house is worth less to him than to a buyer who wants to live in NYC.

Let us now consider a different scenario. We have some financial instrument (e.g., a stock) with some *true "objective" value* (based on the dividends that the stock will generate). Players, however, have *uncertainty* about what the value of the instrument is. To model this, consider now a random variable X on the probability space (Ω, f)—think of X as the value of the financial instrument. (Note that in every state ω, $X(\omega)$ is some fixed value; players, however, are uncertain about this value since they do not know what the true state of the world is.) Assume that one of the players, say i, owns the financial instrument and wants to sell it to player j. When can such trades take place? A natural condition for a trade to take place is that there exists some price p such that j's *expected valuation* for the instrument is at least p and i's expected valuation for the instrument is less than p (otherwise, if i is "risk-neutral", i would prefer to hold on to it.) In other words, we say that the instrument defined by random variable X is **p-tradeable** between i and j at ω if $\mathbb{E}[X \mid P_i(\omega)] < p$ and $\mathbb{E}[X \mid P_j(\omega)] \geq p$. Let $trade_p^X(i,j)$ the event that the instrument (modeled by X) is p-tradeable between i and j.

Using a similar proof to the agreement theorem, we can now obtain a surprising **no-trade theorem**, which shows that it cannot be common knowledge that an instrument is tradeable! The intuition for why this theorem holds is that if I think the instrument is worth $> p$, and then find out that you are willing to sell an instrument at p (and thus must think it is worth $< p$), I should update my beliefs with the new information that you want to sell, which will lower my own valuation.

Theorem 17.7. *Let $M = (n, \Omega, X, s, \vec{P})$ be a knowledge structure, let p be a common prior over Ω, and let X be a random variable over (Ω, p). There does not exists some world $\omega \in \Omega$ and price p such that $\omega \in \mathbf{C}\, trade_p^X(i,j)$*

Proof. Assume for contradiction that there exists some $\omega \in \mathbf{C}\, trade_p^X(i,j)$. Again, let C denote the set of states that are reachable from ω, and let $\omega_1^i, \ldots \omega_{m_i}^i$ denote a set of states such that the beliefs of i in those states "tile" all of C. Since $trade_p^X(i,j)$ holds at every state in C, we have that for every $k \in [m_i]$,

$$\mathbb{E}[X \mid P_j(\omega_k^j)] < p$$

By expanding out the definition of conditional expectation (using Claim A.25),

we get:

$$\mathbb{E}[X \mid C] = \sum_{k \in [m_i]} \mathbb{E}[X \mid P_j(\omega_k^j)] \Pr[P_i(\omega_{k_i}^i) \mid C] < \sum_{k \in [m_i]} p \Pr[P_i(\omega_{k_i}^i) \mid C] = p$$

But, by applying the same argument to player j, we instead get that

$$\mathbb{E}[X \mid C] \geq p$$

which is a contradiction. ■

So, how should we interpret this suprising result? Obviously people are trading financial instruments! We outline a few reasons why such trades may be taking place.

- We are assuming that both players perceive the value of the instrument as the *expectation* of its actual value (where the expectation is taken according to the players' beliefs)—that is, we assume both players are risk-neutral. But, in reality, different players may have different "risk profiles", and may thus be willing to trade even if they have exactly the same beliefs. For instance, an institutional investor (who is risk-neutral) would be very happy to buy an instrument that is worth $\$10M$ with probability $\frac{1}{2}$ and 0 otherwise, for $\$4M$, but a small investor (who is risk-averse) maybe be very happy to agree to sell such an instrument for $\$4M$ (thus, agreeing to take a "loss" in expected profits for a sure gain).

- Another reason we may see trades taking place is that the common knowledge assumption may be too strong. While it is natural to assume that I must know that you want to trade with me in order for the trade to take place, it is not clear that we actually must have common knowledge of the fact that we both agree to the trade; thus trades can take place before common knowledge of the trade taking places "has occurred".

- Finally, and perhaps most importantly, the theory assumes that all players are rational. In reality, there are lots of **noise-traders** whose decision to trade may be "irrational" or "erratic"; noise-traders do not necessarily trade based on their expected valuation of the instrument—in fact, such a valuation may not even be well-defined to them. Indeed, many private persons buying a stock (or index fund) probably have not done enough research to even establish a reasonable belief about the value of the company (or companies in the case of an index fund), but rather buy based on the "speculative" belief that the market will go up.

17.5 Justified True Belief and the Gettier problems.

In our treatment of knowledge and beliefs, we are assuming that the knowledge structure is exogenously given (for instance, in the muddy children example, it was explicitly given as part of the problem description). Sometimes, however, it is not even clear how the right knowledge structure for a situation should be defined. In fact, defining what a player knows is a classic problem in philosophy. The classic way of defining knowledge is through the, so-called, **justified true belief (JTB)** paradigm: according to it, you know ϕ if a) you believe ϕ, b) you have a "reason" to believe it (i.e., the belief is justified), and c) ϕ is actually true.

However, there are several issues with this approach, as demonstrated by the now-famous "Gettier problems": Let us say that you walk into a room with a thermometer. You observe that the thermometer say 70 degrees, and consequently you believe that the temperature is 70 degrees; furthemore, the temperature in the room is 70 degree. According to JTB paradigm, you would then know that the temperature is 70 degrees, since a) you believe it, b) you have a reason for doing so (you read it off the thermometer), and c) the temperature indeed is 70 degrees. Indeed, if the thermometer is working, then this seems reasonable.

But what if, instead, it just so happened that the thermometer was broken but stuck on 70 degree (by a fluke). In this case, would you still "know" that it is 70 degrees in the room? Most people would argue that you do not, but according to the JTB paradigm, you do "know" it. As this discussion shows, coming up with the "right" knowledge structure for a particular situation is thus not always easy and requires careful modeling.

Notes

The muddy children puzzle is from [Lit53]. As mentioned above, our approach to modeling beliefs and knowledge follows that of the philosopher Saul Kripke [Kri59]. A similar approach was introduced independently, but subsequently, by the game-theorist Robert Aumann [Aum76]. (Aumann received the Nobel prize in economics in 2005).

Our treatment most closely follows the works of Kripke [Kri59], Hintikka [Hin62] and Halpern and Moses [HM90]. The analysis of the Muddy Children puzzle is from [HM90]. An extensive treatment of knowledge and common knowledge can be found in [FHMV95].

Aumann's agreement theorem was proven in [Aum76] and the "no-trade theorem" was proven by Milgrom and Stokey [MS82] (although our formalization of this theorem is somewhat different).

The JTB approach is commonly attributed to the works of Plato (from approxiamtely 2400 years ago), but it has also been argued that already Plato pointed out issues with this appraoch; the "Gettier problems" were introduced by the philosopher Gettier [Get63].

Chapter 18

Common Knowledge of Rationality

In this chapter, we turn to using our model of knowledge to reason about games.

18.1 Knowledge and Games

We must first define what it means for a knowledge structure to be "appropriate" for a game G:

Definition 18.1. We say that a knowledge structure tuple $M = (n, \Omega, X, s, \vec{P})$ is **appropriate** for a game $G = (n', A, u)$ if:

- $n' = n$ (the players are the same)
- $X = A$ (the outcome in each world is an action profile in the game)
- For all $\omega \in \Omega, i \in [n]$, and all $\omega' \in P_i(\omega)$, $s_i(\omega') = s_i(\omega)$, where s_i denotes the i^{th} component of s. (That is, players always know their own strategy.)

In other words, each world models what action each of the players take; players have beliefs about what others are doing and believing, and we assume players always know their own action. See Figure 18.1 for a simple knowledge structure appropriate for the Prisoner's Dilemma Game.

We can now define an event RAT_i which denotes the event that player i is acting "rationally". We take a *very weak* definition of what it means to be rational; basically, we define what it means to not be "crazy" or "totally

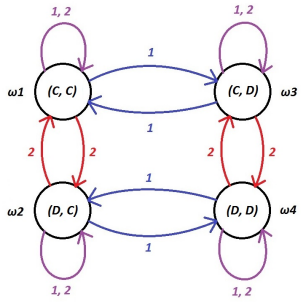


Figure 18.1: A knowledge network for the Prisoner's Dilemma. This graph look similar to the one we consider for the Muddy Children Puzzle, but notice that there each player knows the other player's state but not their own, whereas here each player knows their own strategy but not the other's. Also observe that, while in this network, there is exactly one state per action profile, this does not necessarily need to hold; other networks for the same game may have several nodes with the same action profile but different beliefs.

irrational", by saying that player i is rational at a world state ω if for every alternative strategy a_i' there exists *some world* that i considers possible where he is not strictly better off switching to a_i'. That is, let

$$\mathsf{RAT}_i = \{\omega \in \Omega \mid \forall a_i' \, \exists \, \omega' \in P_i(\omega) : u_i(a_i', s_{-i}(\omega')) \leq u_i(s(\omega'))\},$$

and let RAT denote the event that everyone is rational:

$$\mathsf{RAT} = \bigcap_{i \in [n]} \mathsf{RAT}_i$$

Note that if a player i is rational at some state ω, this is equivalent to saying that there is no action a_i' that strictly dominates $s_i(\omega)$ in every world i considers possible. Based on this observation, we can get the following simple characterizations of states where RAT holds:

Theorem 18.2. *Let $G = (n, A, u)$ be a normal-form game, and let $\vec{a} \in A$ be an action profile. Then the following statements are equivalent:*

1. *$\vec{a}$ is not strictly dominated (i.e. $\vec{a}$ survives one round of iterative strict dominance).*
2. *There exists a knowledge structure $M = (n, \Omega, X, s, \vec{P})$ that is appropriate for G and a state $\omega \in \Omega$ such that $s(\omega) = \vec{a}$ and $\omega \in \mathsf{RAT}$.*

Proof. **[Advanced]** We first prove that (1) implies (2) and then that (2) implies (1).

(1) → (2). Consider an action profile $\vec{b}$ that survives one step of iterative strict dominance. Construct a knowledge structure M where we have a state $\omega_{\vec{a}}$ for every possible action profile $\vec{a}$ such that:

- Let $s(\omega_{\vec{a}}) = \vec{a}$;
- Let $P_i(\omega_{\vec{a}}) = \{\omega_{(a_i, a_{-i})} \mid a_{-i} \in A_{-i}\}$ (that is, i has no idea what any other player is doing, but knows it own action).

It is easily verified that in every world, players know their beliefs and their strategy; they know their strategy by definition, and the fact that they know their beliefs follows from the fact that the beliefs of a player i are determined solely by their strategy. We can now claim that $\omega_{\vec{b}}$ is the world we are looking for. By our definition of M, $s(\omega_{\vec{b}}) = \vec{b}$. Furthermore, since $\vec{b}$ is not strictly dominated, we have by the definition of RAT that $\omega_{\vec{b}} \in \mathsf{RAT}$.

(2) → (1). Assume there exists some knowledge structure M appropriate for G and some state $\omega \in \Omega$ such that $s(\omega) = \vec{a}$ and $\omega \in \mathsf{RAT}$. Assume for contradiction that there exists some player i and some strategy a_i' that strictly dominates a_i. Since $\omega \in \mathsf{RAT}_i$, there must exist some state $\omega' \in P_i(\omega)$ such that:

$$u_i(a_i', s_{-i}(\omega')) \leq u_i(s(\omega'))$$

But, because players "know their own strategy" (since M is appropriate for G), we have that $s_i(\omega') = s_i(\omega) = a_i$. It follows that

$$u_i(a_i', s_{-i}(\omega')) \leq u_i(a_i, s_{-i}(\omega')),$$

which contradicts the fact that a_i' strictly dominates a_i. ■

18.2 An Epistemic Characterization of ISD

In this section, we instead show how to get an "epistemic" (i.e, knowledge-based) characterization of ISD and PNE. As we saw in Chapter 16, it is often natural to not only assume that everyone is rational, but also that it is *commonly known* that everyone is rational (i.e., everyone is rational, everyone knows that everyone is rational, everyone knows that everyone knows that everyone is rational and so on)—this is referred to as *common knowledge of rationality (CKR)*. The following theorem shows that CKR characterizes exactly the set of strategies surviving ISD.

Theorem 18.3. *Let $G = (n, A, u)$ be a normal-form game, and let $\vec{a} \in A$ be an action profile. Then the following statements are equivalent:*

1. *$\vec{a}$ survives iterative strict dominance.*
2. *There exists a knowledge structure $M = (n, \Omega, X, s, \vec{P})$ that is appropriate for G and a state $\omega \in \Omega$ such that $s(\omega) = \vec{a}$ and $\omega \in \boldsymbol{C}\ \mathsf{RAT}$ (i.e., common knowledge of rationality holds at ω).*

Proof. [**Advanced**] Again, we separately show each direction.

(1) → (2). Let ISD_i be the set of strategies for player i surviving ISD, and let $ISD = ISD_1 \times ISD_2 \times \ldots \times ISD_n$ be the set of strategy profiles surviving ISD. We construct a knowledge structure M where we have a state $\omega_{\vec{a}}$ for every possible action profile $\vec{a}$ such that:

- Let $s(\omega_{\vec{a}}) = \vec{a}$;
- Let $P_i(\omega_{\vec{a}}) = \{\omega_{(a_i, a_{-i})} \mid a_{-i} \in ISD_{-i}\}$ (that is, i knows its own action, but otherwise considers all strategy profiles for $-i$ that survive ISD possible).

Just as in the proof of Theorem 18.2, we can easily verify that in every world, players know their strategy and beliefs. We additionally have the following claim:

Claim 18.4. *$\omega \in \mathsf{RAT}$ for every $\omega \in \Omega$.*

Proof. Assume for contradiction that there exists some state ω and some player i such that $\omega \notin \mathsf{RAT}_i$. That is, there exists some a_i' that strictly dominates $a_i = s_i(\omega')$ with respect to $s_{-i}(\omega')$ for all $\omega' \in P_i(\omega)$. By our construction of M (specifically, P), a_i' thus strictly dominates a_i with respect to ISD_{-i}. Thus, intuitively, a_i should have been deleted by ISD (since a_i' dominates it w.r.t the remaining set of strategies), which would be a contradiction. In fact, the only reason why a_i may not be removed in the ISD process at this stage is if a_i' were not inside ISD_i, since ISD only considers strict dominance by strategies still surviving. But in that case, a_i' was deleted due to being dominated by some strategy a_i^1; in turn, this strategy is either in ISD_i or was previously removed due to being dominated by some strategy a_i^2. Since the strategy space is finite, we must eventually will reach some $a_i^m \in ISD_i$, which strictly dominates a_i itself with respect to ISD_{-i} by transitivity of strict dominance (and the fact that the strategy space shrinks, preventing

a strategy strictly dominated earlier in the process from not being strictly dominated later). This, then, contradicts the assumption that $a_i \in ISD_i$. ■

Given this claim, it directly follows that for every player i, and state $\omega \in \Omega$, $\omega \in \mathbf{K}_i$ RAT; thus, $\omega \in \mathbf{E}$ RAT; consequently, we thus have that for every $\omega \in \Omega$,$\omega \in \mathbf{E}\ \mathbf{E}$ RAT, and so on. By induction it follows that for every $\omega \in \Omega$, and every k, $\omega \in \mathbf{E}^k$ RAT, and thus we also have that $\omega \in \mathbf{C}$ RAT.

Thus for every strategy profile $\vec{a}$ that survives ISD, we have that M and the state $\omega_{\vec{a}}$ satisfy the required conditions.

(2) → (1). Consider some knowledge structure M appropriate for G. Let ISD_i^k denote the set of strategies for player i surviving k rounds of ISD (and define ISD^k as the set of strategy profiles surviving ISD). We shall prove by induction that for any state $\omega \in \mathbf{E}^k$ RAT (i.e., where everybody knows that everybody knows ...(k times) ...that everyone is rational), we have $s(\omega) \in ISD^k$. The base case ($k = 0$) is proven by Theorem 18.3 above.

For the inductive step, assume the statement is true for k, and let us prove it for $k + 1$. Consider some $\omega \in \mathbf{E}^{k+1}$ RAT and some player i. Note that if $\omega \in \mathbf{E}^{k+1}$ RAT, then $\omega \in \mathbf{E}^k$ RAT (since, by the definition of beliefs, $\omega \in P_i(\omega)$); consequently, $\omega \in$ RAT as well. So, by the induction hypothesis, $s_i(\omega) \in ISD_i^k$; furthermore, for every $\omega' \in P_i(\omega)$, $s_{-i}(\omega') \in ISD_{-i}^k$. Since $\omega \in$ RAT, it follows by definition that $s_i(\omega)$ is an action that is inside ISD_i^k, but not strictly dominated by any action a_i' with respect to ISD_{-i}^k; hence $s_i(\omega) \in ISD_i^{k+1}$ (i.e., it will survive one more round). And since the above holds for all players, we have $s(\omega) \in ISD^{k+1}$. So, for any $\omega \in \mathbf{C}$ RAT, $s(\omega) \in ISD^k$ for any k, implying that $s(\omega)$ survives ISD. ■

18.3 An Epistemic Characterization of PNE

Can we hope to also get an epistemic characterization of PNEs? To do this, we need to add additional assumptions about the knowledge of players. In particular, we will need to assume that everyone knows not only their own strategy, but also the strategies of others! Specifically, let KS be the event that all players know all players' strategies; formally,

$$\mathsf{KS} = \{\omega \in \Omega \mid \forall i \in [n], \omega' \in P_i(\omega) : s(\omega') = s(\omega)\}$$

Theorem 18.5. *Let $G = (n, A, u)$ be a normal-form game, and let $\vec{a} \in A$ be an action profile. Then the following statements are equivalent:*

1. *$\vec{a}$ is a PNE.*

2. *There exists a knowledge structure $M = (n, \Omega, X, s, \vec{P})$ that is appropriate for G and a state $\omega \in \Omega$ such that $s(\omega) = \vec{a}$ and $\omega \in \mathsf{KS} \cap \boldsymbol{C}\ \mathsf{RAT}$.*

3. *There exists a knowledge structure $M = (n, \Omega, X, s, \vec{P})$ that is appropriate for G and a state $\omega \in \Omega$ such that $s(\omega) = \vec{a}$ and $\omega \in \mathsf{KS} \cap \mathsf{RAT}$.*

Proof. **[Advanced]** We prove equivalence by showing that (1) implies (2) implies (3) implies (1).

(1) → (2). Consider a structure M with just one state ω such that $s(\omega) = \vec{a}$ and $P_i(\omega) = \omega$. Clearly, at ω, KS and **RAT** hold, and consequently, by induction, also **C** RAT.

(2) → (3). Trivial.

(3) → (1). KS and RAT taken together imply that for every player i, player i's strategy at ω is a best response w.r.t. $s_i(\omega)$; thus, $s(\omega) = \vec{a}$ is a PNE. ■

18.4 Knowledge vs. Belief: Explaining Bubbles in the Market

So far in our discussion we have only considered a model of knowledge. We may also use a similar model to reason about belief; we can define a *belief structure* in exactly the same way as a knowledge structure, except that we remove the second to last condition in Definition 17.2—that is, we no longer require that in every state ω, players always need to consider ω possible. We say that a player i believes ϕ if ϕ holds in every world i considers possible, and we define a belief event $\mathbf{B}_i$ in exactly the same ways as $\mathbf{K}_i$ (except that it is defined on a "belief structure" rather than a "knowledge structure").

To see how this makes things different, let us once more consider an auction scenario with two players and a single item and the knowledge structure depicted in Figure 18.2. Assume that the true state of the world is ω_1. Player 1 knows the true state of the world. However, player 2 has an incorrect belief: he believes that ω_2 is the true state of the world. (Notice that there is no self-loop for player 2 at ω_1, so he does not even consider the true state possible; this could not have happened in a knowledge structure.) Thus, player 2 believes that player 1 believes that player 2 values the item at 100, when player 1 (correctly) believes that player 2 values it at zero!

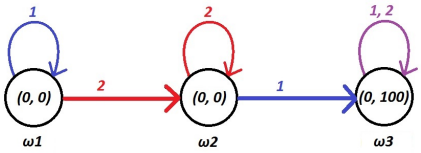


Figure 18.2: A belief network for a second-price auction. Some states in a belief network (here, ω_1 and ω_2) lead to players having incorrect beliefs.

In other words, player 2 (incorrectly) believes that he is smarter than player 1, who (according to player 2) incorrectly believes that player 2 is foolishly valuing the good at 100. How much do you think an auctioneer could sell this item for in such a situation? Remember that not only everyone thinks that the item is worthless, but we also have that everyone believes that everyone thinks the item is worthless! But, as is turns out, just the fact that somebody believes that *somebody else believes* that the item is worth 100 means that a sufficiently clever auctioneer can sell it for close to 100, assuming that *common belief of rationality* holds (i.e. it is common belief that everyone is rational, where common beliefs is defined exaclty as common knowledge but in belief structures). In essence, we get a "bubble" in the market where the item gets sold at a high price despite nobody actually having a high valuation for it.

We will briefly and informally explain how this can happen. Consider a standard second-price auction, but change it slightly so that the seller has some small reward $R < 1$ that he gives back to the players as a "participation gift", but splits based on how much the players bid. Intuitively, adding such a gift should make it worse for the seller, as he now loses R out of the profit he makes from selling the item. But, as we shall see (and as is common in practice), adding small "welcome gifts" can be a way to extort more money from the buyers! More precisely, in an auction with n players, player i gets $R\frac{b_i}{n(1+b_i)}$ as a participation gift where b_i is the player's bid; note that each such reward is smaller than $\frac{R}{n}$ and thus in total, the auctioneer never spends more than R. Intuitively, however, this reward incentivizes players to bid high, especially if they are sure to never win in the auction (as in this case, they increase their participation reward). As we shall argue, in *any* outcome that is consistent with CKR, the object is sold for close to 100. The reasoning works as follows:

- We first claim that at ω_3, a rational player 2 must bid *at least* 100. By bidding less, he can never get a higher utility in the second-price auction (since he will anyways end up paying the same price if he wins the auctions, and in case he looses, we get a utility of 0); furthemore, by bidding less, he gets a strictly smaller part of the reward R. Thus we conclude that player 2 gets strictly by bidding less than 100, no matter what the other player does.

- Now, in ω_2, since player 1 believes (incorrectly) that the true state is ω_3, a rational player 1 must bid at least 99: he knows that player 2 (being rational) bids at least 100, thus by bidding 99 he gets as much as possible of the participation reward.

- Thus, in the true state of the world, ω_1, player 2, who incorrectly believes that the true state is ω_2, must bid at least 98 (since he is sure that player 1 will bid at least 99).

- Finally, at ω_1, player 1, who correctly believes that the true state is ω_1 must bid at least 97 (since he is sure that player 2 will bid at least 98.

We conclude that true state of the world, ω_1, player 1 will bid at least 97 and player 2 will bid at least 98; thus, the item will be sold for at least 97, and the auctioneer is guaranteed to receive a utility of at least $97 - R \geq 96$ for a worthless item!

Notes

Our treatment of the knowledge-based characterizations of ISD and PNE follows ideas from [TW88, AB95, HP09], but we explore them here in a somewhat different setting which enables a significantly simpler analysis. As far as we know, these characterizations are new (and our formalism is closest in spirit to [HP09, CMP16]).

Closely related characterization results was first proven by [TW88] and [AB95] relying on a diffent (stronger) notion of rationality. More precisely, in our treatment we consider a weak "possibilistic" notion of rationality. A stronger definition of rationality can be obtained by considering knowledge structures with probabilities: we no longer only have just a set of worlds players consider possible at each world, but we also assign probabilities to the worlds a player considers possible (e.g., a player may consider some worlds more likely than others). In such a setting, the most popular notion of rationality requires a player i's strategy at ω to *maximize i expected utility* given

the distribution of strategies for players $-i$ induced by their beliefs. CKR is characterized by ISD in this setting as long as we change the definition of ISD to also allow for domination by "mixed strategies" (that is, we remove actions that are strictly dominated by a probability distribution over actions)—this was the result first proved by [TW88]. An analogous characterizations of (mixed-strategy) Nash equilibria were obtained in [AB95].

The bubbles in the market example originates from the treatment in [CMP16] where a very similar auction mechanism is analyzed (but the actual example is new).

Chapter 19

Markets with Network Effects

Recall our model of networked coordination games (from Chapter 4), where the utility of a player's action depended on a) the "intrinsic value" of the action to the player, and b) the coordination ("network") value of the action, which was a function of the number of its neighbors choosing it.

In this chapter, we will now study markets in a similar setting. Whereas we previously studied markets in a setting where the number of goods for sale was limited (i.e. matching markets), we here consider a market where we have an unlimited number of copies of the good for sale, but buyers still are only interested in buying one item. As we shall see, player's beliefs (and even their higher-level beliefs) will be instrumental for analyzing such scenarios.

19.1 Simple Networked Markets

Concretely, let us consider a scenario where a good can be mass produced at some cost p, and assume that the good is being offered for sale at the same price p; this models the idea that if there are multiple producers, competition among them will drive the price of the good down to $p + \epsilon$ for some small ϵ.[1]

A Market without Network Effects First, consider a simple market with just an *intrinsic value* for the good, but assume players have different intrinsic values—formally, these values are modeled as player types. Let us, further, assume we have a *large set of buyers*—let $F(p)$ denote the fraction of the

[1] At this point, the astute reader may ask: why would anyone ever want to produce a good if they only recover their production cost? The point is that the production cost here includes also the "cost of financing"—e.g., whatever revenue shareholders of the company producing the good are expecting to get.

buyers whose intrinsic value t for the good is at least p. As we did in Chapter 10, let us consider a quasi-linear utility model, where a buyer's utility for buying an item is their value for the item minus the price they pay for it. In such a utility mode, a buyer with type t who is rational will only be willing to buy the good if its price p is at most their type, t. Thus, if the good is priced at p^*, then assuming all buyers are rational, a $F(p^*)$ fraction of the buyers will buy it.

Modeling Network Effects Let us now modify the model by introducing network effects. Towards this end, let us define the value of a good to a player with type (i.e., intrinsic value) t as

$$v = td(x)$$

where $d(x) > 0$ is a "multiplier" modeling the strength of the network effect and x is the fraction of the population that buys the good. We here restrict our study to monotonically increasing multipliers $d(x)$, so that if more users have the good, the good becomes more valuable (to everyone). (We point out that it may also make sense in some situations to consider a "negative network effect" such that the good becomes *less* valuable when too many players have it—as the famous Yogi Berra quote goes: "Nobody goes there anymore. It's too crowded.")

Analyzing Markets with Network Effects: Self-fulfilling Beliefs In such a model, what fraction of the buyers will buy a good? To answer this question, we need to consider the beliefs of the players.

- If a buyer with type t believes that an x fraction of the population will buy the good, then their *believed value* for the good is $v = td(x)$.
- Thus, if the good is priced at p^*, a rational buyer, wishing to maximize their *expected utility*, will only agree to buy the good if $t \geq \frac{p^*}{d(x)}$.
- We conclude that If *everyone* is rational and believes that an x fraction of the population will buy the good, then an $F\left(\frac{p^*}{d(x)}\right)$ fraction of buyers will actually buy.

Of course, beliefs may be incorrect. So, if people initially believe that, say, an $x_0 = 0$ fraction of the players will buy the good, then $x_1 = F\left(\frac{p^*}{d(x_0)}\right)$ will buy it; but, then, given this, $x_2 = F\left(\frac{p^*}{d(x_1)}\right)$ should actually buy it, and so on and

so forth—in essence, this iteration gives rise to a **best-response dynamic for markets with network effects**.

Consequently, following the general theme of this course, we are interested in the "fixed-points", or equilibria, of this system, where buyers' beliefs are correct: That is, situations where if everyone is rational and believes that an x^* fraction will buy the good, then an x^* fraction will actually buy it; we refer to these as *self-fulfilling beliefs*. More precisely, we refer to a belief x^* as a **self-fulfilling (w.r.t., F, p^*)** if

$$F\left(\frac{p^*}{d(x^*)}\right) = x^*$$

We will also call such an x^* an **equilibrium**.

Stable and Unstable Equilibria To better understand the structure of self-fulfilling beliefs, let us consider an example. Assume $F(p) = \max(1-p, 0)$. (So nobody has intrinsic value higher than 1, but everyone has a non-negative intrinsic value.) Let $d(x) = a + bx$; we will focus on situations where a is small (so that the value of the good is tiny if nobody else has it) but b is large (so that the value is heavily dependent on the network effect). Self-fulfilling beliefs are characterized by the equation,

$$x = F\left(\frac{p^*}{d(x)}\right) = \max\left(1 - \frac{p^*}{a+bx}, 0\right)$$

So, $x = 0$ is a self-fulfilling belief if and only if $1 - \frac{p^*}{a} \leq 0$; that is, when $p^* \geq a$. Any other self-fulfilling belief must an x such that $0 < x \leq 1$ such that:

$$1 - \frac{p^*}{a+bx} = x$$

which simplifies to:

$$a + bx - p^* = ax + bx^2$$

$$x^2 + \frac{a-b}{b}x + \frac{p^* - a}{b} = 0$$

By solving the quadratic equation, we get the solutions:

$$x = \frac{b-a}{2b} \pm \sqrt{\frac{a-p^*}{b} + \left(\frac{a-b}{2b}\right)^2}$$

Thus, as we see, there can be between 0 and 3 equilibria, depending on our choice of a, b, and p^*. For instance, setting $a = 0.1$, $b = 1$, $p^* = 0.2$ produces

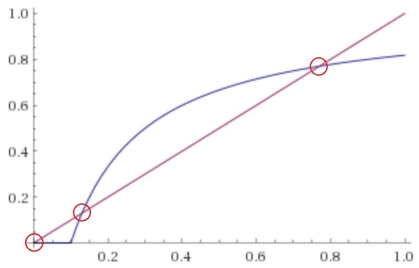


Figure 19.1: The function $y = F\left(\frac{p^*}{d(x)}\right)$ for $(a, b, p^*) = (0.1, 1, 0.2)$ is plotted in blue, and the function $y = x$ is plotted in red. Equilibria (w.r.t F, p^*) are thus characterized by the intersections between plots . Also, notice that BRD will converge to a higher value when $F\left(\frac{p^*}{d(x)}\right) > x_1^*$ and lower when $F\left(\frac{p^*}{d(x)}\right) < x_1^*$, which shows that x_1^* is an unstable equilibrium.

the example shown in Figure 19.1. $x_0^* = 0$ is a self-fulfilling belief, since $p^* > a$ (if everybody believe that nobody will buy the good, the value of the good becomes too low for all the player and thus nobody will want to buy it). The other two equilibria are given by the solutions to the quadratic equation: $x_1^* \approx 0.13$ and $x_2^* \approx 0.77$.

Notice that these three outcomes are actually very different. Which of them will we arrive at assuming that people follow the best-response dynamics considered above? Clearly, if we start at an equilibrium, the BRD process will not move away from it. But what if players' beliefs are just a touch off (i.e., we start the BRD at a point close to the equilibrium), will we always converge to the equilibrium point? The answer turns out to be no: In fact, if we start anywhere between x_0^* and x_1^*, BRD will converge to x_0^*, no matter how close to x_1^* we start! Similarly, if we start at any point greater than x_1^*, no matter how close to x_1^*, BRD will converge to x_2^*. We thus call x_0^* and x_2^* **stable equilibria**; x_1^*, on the other hand is an **unstable equilibrium**.

Tipping Points Notice, however, that the equilibrium point x_1^* is still interesting as a, so-called, **tipping point**: If everyone believes that more than an x_1^* fraction of the players will buy the good, we end up at the "good" equilibrium x_2^* where almost everyone buys it; if people believe that a fewer than an x_1^* fraction of the players will buy the good, we in turn end up at the "bad" equilibrium x_0^* where *nobody* buys it!

So, in order to be successful at selling their good, the seller needs to make sure that they achieve a penetration rate of at least x_1^*. In order to succeed at this, they can lower the price p^* of the good to make the "tipping point" lower still. If they lower p^* to 0.15, for instance, $x_1^* \approx 0.06$; at 0.11, $x_1^* \approx 0.01$! Thus, to successfully market their good, a seller should be willing to lower the price of their good *even below its production cost* until they achieve some appropriate penetration rate (above the corresponding "tipping point" to its actual production cost). At that point, they can then safely increase the price back to their production cost.

19.2 Markets with Asymmetric Information

Self-fulfilling beliefs are also a useful tool to analyze markets with *asymmetric information*—that is, markets where the seller might have more information about the good than the buyers.

The Market for Lemons Consider a simple market for used cars. We have a large set of cars owned by some sellers:

- A fraction f of these cars are *"good"*, worth v_B to buyers but $v_S < v_B$ to the sellers (note that we assume that *all* buyers and sellers have the same valuation for a good car).
- The remaining cars are *"lemons"*, worthless to both buyers and the sellers alike.

Sellers may decide to keep their car, or decide to put it up for sale (i.e. placing their car on the market).

How much would a buyer be willing to pay for a *random* car *on the market* (i.e., a random car that has been put up for sale), assuming that lemons and good cars are difficult to distinguish. If using the quasi-linear utility model, a *rational* buyer wishing to maximize their expected utility will only be willing to pay at most

$$v_B x$$

if they *believe* that an x fraction of the cars *on the market* are good.

So, what should a seller do?

- If the seller has a lemon, they should clearly put it up for sale, since they might be able to get some money for it.

- On the other hand, when the seller has a good car and *believes that buyers believe* that an x fraction of the cars on the market are good, then if

$$v_B x \geq v_S$$

they would be willing to put it up for sale, and otherwise they would not (since they will never be able to get more than v_S, so no point putting it up for sale).

In fact, to make this last argument, we additionally must assume that *common knowledge of rationality* holds so that we assume that the seller is rational and believes that all buyers are rational (and thus only buy for at most $v_B x$).

Analyzing Self-fulfilling Beliefs: Market Crashes To analyze what happens in such a market, we will again consider self-fulfilling beliefs: We say that the belief x^* is **self-fullfilling (w.r.t.** v_s, v_b) if *common knowledge of rationality* and *common knowledge that "a fraction x^* of the cars on the market are good"* implies that an x^* fraction of the cars on the market will be good.

Note that common knowledge of "a fraction x^* of the cars on the market are good" means that sellers believes buyers believe an x^* fraction of the cars on the market are good. Thus, all sellers believe that all buyers will agree to buy cars for $v_B x^*$ (but no less). Since all sellers have the same valuation v_S for their car, either all of them will put it up for sale, or none. Thus, the only possible self-fulfilling equilibria are:

- $x^* = f$ (when all sellers put their car up for sale); or,
- $x^* = 0$ (when no sellers of good cars put them put it up for sale).

The "good" equilibrium when all sellers put their car up for sale, can only happen when $v_S \leq v_B f$, that is when the total fraction f of good cars is at least $\frac{v_S}{v_B}$. On the other hand, when $f < \frac{v_S}{v_B}$, then the only self-fullfilling equilibrium is $x^* = 0$, where nobody puts good cars up for sale—we get a "market crash" where only sellers with lemons put them up for sale, and nobody is willing to buy!

In fact, the situation is even worse: Even if the original fraction of good cars is high (i.e., $f > \frac{v_S}{v_B}$), if buyers believe that an $x < \frac{v_S}{v_B}$ fraction of cars are on the market are good, then none of them will be willing to buy; as a consequence, sellers of good cars will no longer be willlng to sell. In other words, running a best-response dynamics (as above) starting off at some $x < \frac{v_S}{v_B}$ will lead to the "bad" equilibrium $x^* = 0$ (in just one step of the

dynamics). On the other hand, if we start off at some $x > \frac{v_S}{v_B}$ we instead converge to $x^* = \frac{v_S}{v_B}$ (again, in just one step).

We conclude that $t = \frac{v_S}{v_B}$ is the "tipping point" for the market: if *the sellers believe that the buyers believe* that the fraction of good cars on the market is smaller than $\frac{v_S}{v_B}$, they will not be willing to sell good cars, and the market crashes (as before, only lemons are put on the market, and nobody will be willing to buy them).

Of course, it is unrealistic to assume that every good car has the same value. But even if there are several different types of cars (e.g., "excellent", "very good", "good", "bad", and "lemons"), but different quality cars are indistinguishable to buyers, if buyers' beliefs on the "quality of the market" falls below some appropriately-defined tipping point, none of the "excellent" cars will enter the market, which will drive down the expected value of a random car and consequently drive out all the "very good" ones, which will drive out the "good" ones, etc, until only lemons remain in the market! So we get the same type of market crash also in more realistic models. (Indeed, 5 years after Akerlof's seminal paper on markets for lemons [Ake95] was published, the US enacted federal "lemon laws" to protect buyers against purchases of lemons.)

Extension to Labor Markets Finally, let us mention that this sort of market crash occurs not only in used car sales but also in, for instance, labor markets where an employer might not be able to effectively evaluate the skills of a worker. Consider, for instance, a simple seeting where we have 2 types of workers ("good workers" and "bad workers"): Good workers are analogous to the good cars, and have some value v_S they can generate on their own (e.g. being self-employed) and some other value v_B they can contribute to the employer; bad workers are the "lemons" of the market (worth nothing on their own and to the employer). The exact same analysis as above applies to this situation assuming the employer cannot distinguish the two types of worker.

Spence's signalling model. *(Or: why university degrees are useful, even if you don't actually learn anything!)* One way around this problem is to let players *signal* the value of their good. For the labor market, education can be used as such a signal. Assume that getting a university degree is more costly for a bad worker than for a good one (due to, say, fewer scholarships, more time required, etc.). If companies set different base salaries depending on whether the worker has a degree or not, they may end up at a "truthful equilibrium" where bad workers do not get degrees because the added salary is not worth the extra cost of the degree to them, while for good workers,

getting the degree will be worthwhile, since the extra salary outweighs the (lower) cost of the education. So, getting a degree may be worthwhile to the good workers—*even if they do not gain any relevant skills or knowledge*—as their willingness to get the degree signals their "higher value" as an employee.

Notes

Our treatment of markets with network effects follows models from [KS85, SV13] (see also the treatment in [EK10]). Markets for lemons were first explored in the work of Akerlof [Ake95], and "Spence's signalling model" in the work of Spence [Spe73]. Andrew Spence, George Akerlof, and Joseph Stiglitz shared the 2001 Nobel Prize in Economics their work on markets with asymmetric information.

Bibliography

[Ake95] George Akerlof. The market for lemons: Quality uncertainty and the market mechanism. In *Essential Readings in Economics*, pages 175–188. Springer, 1995.

[ADK+08] Elliot Anshelevich, Anirban Dasgupta, Jon Kleinberg, Eva Tardos, Tom Wexler, and Tim Roughgarden. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing*, 38(4):1602–1623, 2008.

[Arr50] Kenneth J Arrow. A difficulty in the concept of social welfare. *The Journal of Political Economy*, pages 328–346, 1950.

[Aum76] R. J. Aumann. Agreeing to disagree. *Annals of Statistics*, 4(6):1236–1239, 1976.

[AB95] R. J. Aumann and A. Brandenburger. Epistemic conditions for Nash equilibrium. *Econometrica*, 63(5):1161–1180, 1995.

[Ban92] Abhijit V Banerjee. A simple model of herd behavior. *The Quarterly Journal of Economics*, pages 797–817, 1992.

[BA99] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[BP90] Salvador Barbera and Bezalel Peleg. Strategy-proof voting schemes with continuous preferences. *Social choice and welfare*, 7(1):31–38, 1990.

[Bas94] Kaushik Basu. The traveler's dilemma: Paradoxes of rationality in game theory. *The American Economic Review*, 84(2):391–395, 1994.

[BCN05] T. Becker, M. Carter, and J. Naeve. Experts playing the Traveler's Dilemma. Discussion paper 252/2005, Universität Hohenheim, 2005.

[Bla48] Duncan Black. On the rationale of group decision-making. *The Journal of Political Economy*, pages 23–34, 1948.

[Bol84] Béla Bollobás. The evolution of random graphs. *Transactions of the American Mathematical Society*, 286(1):257–274, 1984.

[Bra68] Dietrich Braess. Über ein paradoxon aus der verkehrsplanung. *Mathematical Methods of Operations Research*, 12(1):258–268, 1968.

[CGGH99] M. Capra, J. K. Goeree, R. Gomez, and C. A. Holt. Anamolous behavior in a traveler's dilemma. *American Economic Review*, 89(3):678–690, 1999.

[CIL12] Shuchi Chawla, Nicole Immorlica, and Brendan Lucier. On the limits of black-box reductions in mechanism design. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 435–448. ACM, 2012.

[CMP16] Jing Chen, Silvio Micali, and Rafael Pass. Tight revenue bounds with possibilistic beliefs and level-k rationality. *Econometrica*, 83(4):1619–1639, 2016.

[Che52] Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, pages 493–507, 1952.

[Cla71] Edward H Clarke. Multipart pricing of public goods. *Public choice*, 11(1):17–33, 1971.

[CY92] K. S. Cook and T. Yamagishi. Power in exchange networks: A power-dependence formulation. *Social Networks*, 14:245–265, 1992.

[CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.

[dC94] Jean-Antoine-Nicolas de Caritat marquis de Condorcet. *Essai sur l'application de l'analyse la probabilit des dcisions rendues la pluralit des voix*. De l'Imprimerie royale,, Cambridge, Mass., 1994.

[DGS86] Gabrielle Demange, David Gale, and Marilda Sotomayor. Multi-item auctions. *The Journal of Political Economy*, 94:863–872, 1986.

[DF81] Lester E Dubins and David A Freedman. Machiavelli and the gale-shapley algorithm. *The American Mathematical Monthly*, 88(7):485–494, 1981.

[EK10] David Easley and Jon Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, New York, NY, USA, 2010.

[EOS07] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *The American economic review*, 97(1):242–259, 2007.

[EK72] Jack Edmonds and Richard M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, 1972.

[ER59] P Erdős and A Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[FHMV95] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. MIT Press, Cambridge, Mass., 1995.

[FF62] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[GS62] David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[GS85] David Gale and Marilda Sotomayor. Some remarks on the stable matching problem. *Discrete Applied Mathematics*, 11(3):223–232, 1985.

[Gal07] Francis Galton. Vox populi. *Nature*, 75(1949):450–451, 1907.

[Get63] E. Gettier. Is justified true belief knowledge? *Analysis*, 23:121–123, 1963.

[Gib73] Allan Gibbard. Manipulation of voting schemes: a general result. *Econometrica: journal of the Econometric Society*, pages 587–601, 1973.

[Gib77] Allan Gibbard. Manipulation of schemes that mix voting with chance. *Econometrica: Journal of the Econometric Society*, pages 665–681, 1977.

[Gil59] Donald B Gillies. Solutions to general non-zero-sum games. *Contributions to the Theory of Games*, 4(40):47–85, 1959.

[Gro73] Theodore Groves. Incentives in teams. *Econometrica*, 41(4):617–631, 1973.

[Hal35] P. Hall. On representatives of subsets. *Journal of the London Mathematical Society*, s1-10(1):26–30, 1935.

[HM90] J. Y. Halpern and Y. Moses. Knowledge and common knowledge in a distributed environment. *Journal of the ACM*, 37(3):549–587, 1990.

[HP09] J. Y. Halpern and R. Pass. A logical characterization of iterated admissibility. In *Theoretical Aspects of Rationality and Knowledge: Proc. Twelfth Conference (TARK 2009)*, pages 146–155, 2009.

[HP10] Joseph Y. Halpern and Rafael Pass. Game theory with costly computation. In *Proc. of the First Symposium on Innovations in Computer Science*, pages 120–142, 2010.

[Hin62] J. Hintikka. *Knowledge and Belief*. Cornell University Press, Ithaca, N.Y., 1962.

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[Hur60] Leonard Hurwicz. Optimality and informational efficiency in resource allocation processes. In K.J. Arrow, S. Karlin, and P. Suppes, editors, *Mathematical Methods in the Social Sciences*, pages 27–46. Stanford University Press, Stanford, CA, 1960.

[HZ79] Aanund Hylland and Richard Zeckhauser. The efficient allocation of individuals to positions. *Journal of Political economy*, 87(2):293–314, 1979.

[Kak41] S Kakutani. A generalization of brouwer's fixed point theorem. *Duke Mathematical Journal*, 8:457–459, 1941.

[KS85] Michael L Katz and Carl Shapiro. Network externalities, competition, and compatibility. *The American economic review*, 75(3):424–440, 1985.

[KKT03] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[KT05] Jon Kleinberg and Eva Tardos. *Algorithm Design*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[KT08] Jon Kleinberg and Éva Tardos. Balanced outcomes in social exchange networks. In *Proceedings of the fortieth annual ACM symposium on Theory of Computing*, pages 295–304, 2008.

[Kle99] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[Kle00] Jon M. Kleinberg. Navigation in a small world. *Nature*, 406:845–845, 2000.

[KP09] Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. *Computer Science Review*, 3(2):65 – 69, 2009.

[Kri59] S. Kripke. A completeness theorem in modal logic. *Journal of Symbolic Logic*, 24:1–14, 1959.

[LLP15] Samantha Leung, Edward Lui, and Rafael Pass. Voting with coarse beliefs. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 61–61. ACM, 2015.

[Lit53] J. E. Littlewood. *A mathematician's miscellany*. Methuen, 1953.

[McL11] Andrew McLennan. Manipulation in elections with uncertain preferences. *Journal of Mathematical Economics*, 47(3):370–375, 2011.

[Mil67] Stanley Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967.

[MS82] Paul Milgrom and Nancy Stokey. Information, trade and common knowledge. *Journal of economic theory*, 26(1):17–27, 1982.

[MS96] Dov Monderer and Lloyd S Shapley. Potential games. *Games and economic behavior*, 14(1):124–143, 1996.

[MKA14] Carey K Morewedge, Tamar Krishnamurti, and Dan Ariely. Focused on fairness: Alcohol intoxication increases the costly rejection of inequitable rewards. *Journal of Experimental Social Psychology*, 50:15–20, 2014.

[MP16] Andrew Morgan and Rafael Pass. The price of stability in networked coordination games. In preparation, 2016.

[Mor00] Stephen Morris. Contagion. *The Review of Economic Studies*, 67(1):57–78, 2000.

[Mye81] Roger B Myerson. Optimal auction design. *Mathematics of operations research*, 6(1):58–73, 1981.

[Nas50a] J. Nash. The bargaining problem. *Econometrica*, 18:155–162, 1950.

[Nas50b] J. Nash. Equilibrium points in n-person games. *Proc. National Academy of Sciences*, 36:48–49, 1950.

[Neu28] J v Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928.

[NR99] Noam Nisan and Amir Ronen. Algorithmic mechanism design. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 129–140. ACM, 1999.

[NRTV07] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. *Algorithmic game theory*, volume 1. Cambridge University Press Cambridge, 2007.

[OR94] Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, Cambridge, Mass., 1994.

[PBMW98] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172, Brisbane, Australia, 1998.

[PT10] R. Pass and D. Tseng. A course in discrete structures. Lecture Notes for CS 2800, 2010.

[PS14] Rafael Pass and Karn Seth. On the impossibility of black-box transformations in mechanism design. In *Algorithmic Game Theory: 7th International Symposium, SAGT 2014, Haifa, Israel, September 30–October 2, 2014, Proceedings*, volume 8768, page 279. Springer, 2014.

[Roc84] Sharon C Rochford. Symmetrically pairwise-bargained allocations in an assignment market. *Journal of Economic Theory*, 34(2):262–281, 1984.

[Ros73] R. W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.

[Rot82] Alvin E Roth. The economics of matching: Stability and incentives. *Mathematics of operations research*, 7(4):617–628, 1982.

[RS92] Alvin E Roth and Marilda A Oliveira Sotomayor. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Cambridge University Press, 1992.

[RT02] T. Roughgarden and É. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236–259, 2002.

[SS81] Mark A Satterthwaite and Hugo Sonnenschein. Strategy-proof allocation mechanisms at differentiable points. *The Review of Economic Studies*, 48(4):587–597, 1981.

[Sat75] Mark Allen Satterthwaite. Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social we lfare functions. *Journal of economic theory*, 10(2):187–217, 1975.

[SM03] Andreas S. Schulz and Nicolás E. Stier Moses. On the performance of user equilibria in traffic networks. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms,*

January 12-14, 2003, Baltimore, Maryland, USA., pages 86–87, 2003.

[SV13] Carl Shapiro and Hal R Varian. *Information rules: a strategic guide to the network economy*. Harvard Business Press, 2013.

[SS71] L. S. Shapley and M. Shubik. The assignment game i: The core. *International Journal of Game Theory*, 1(1):111–130, 1971.

[SS74] Lloyd Shapley and Herbert Scarf. On cores and indivisibility. *Journal of mathematical economics*, 1(1):23–37, 1974.

[Spe73] Michael Spence. Job market signaling. *The quarterly journal of Economics*, pages 355–374, 1973.

[Sur05] James Surowiecki. *The wisdom of crowds*. Anchor, 2005.

[Sve99] Lars-Gunnar Svensson. Strategy-proof allocation of indivisible goods. *Social Choice and Welfare*, 16(4):557–567, 1999.

[SR14] Lars-Gunnar Svensson and Alexander Reffgen. The proof of the gibbard–satterthwaite theorem revisited. *Journal of Mathematical Economics*, 55:11 – 14, 2014.

[TW88] T. C.-C. Tan and S. R. da C. Werlang. The Bayesian foundation of solution concepts of games. *Journal of Economic Theory*, 45:370–391, 1988.

[Vic61] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16(1):8–37, 1961.

[vM47] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, N.J., 2nd edition, 1947.

[Wal54] Leon Walras. *Elements of pure economics, or, The theory of social wealth / Leon Walras ; translated by William Jaffe*. Published for the American Economic Association and the Royal Economic Society by Allen and Unwin London, 1954.

[WS98] Duncan J Watts and Steven H Strogatz. Collective dynamics of small-worldnetworks. *Nature*, 393(6684):440–442, 1998.

[Wil99] D. Willer. *Network Exchange Theory*. Praeger, 1999.

[You02] H. Peyton Young. The diffusion of innovations in social networks. Santa Fe Institute Working Paper 02-04-018, 2002.

Part V

Appendix

Appendix A

A Primer on Probability

In this section, we provide a brief introduction to some basic probability theory; our goal is to provide the minimal preliminaries needed to understand the material in the course. The text is taked almost verbatim from [PT10].

Originally motivated by gambling, the study of probability is now fundamental to a wide variety of subjects, including social behavior (e.g., economics and game theory) and physical laws (e.g., quantum mechanics and radioactive decay). But what is probability? What does it mean that that a fair coin toss comes up heads with probability 50%? One interpretation is *frequentist* "50%" means that if we toss the coin 10 million times, it will come up heads in roughly 5 million tosses. A different interpretation is *Bayesian* (or subjective): "50%" is a statement about our beliefs, and how much we are willing to bet on *one* coin toss. For the purpose of this course, the second interpretation will be more relevant to us.

A.1 Probability Spaces

In our treatment, we restrict our attention to *discrete* probability spaces:[1]

Definition A.1 (Probability Space)**.** A **probability space** is a pair (S, f) where S is a countable set called the **sample space**, and $f : S \to [0, 1]$ is called the **probability mass function**.[2] Additionally, f satisfies the property $\sum_{x \in S} f(x) = 1$.

[1] Without formally defining this term, we refer to random processes whose outcomes are discrete, such as dice rolls, as opposed to picking a uniformly random real number from zero to one.

[2] By $[0, 1]$ we mean the real interval $\{x \mid 0 \leq x \leq 1\}$

Intuitively, the sample space S corresponds to the set of possible states that the world could be in, and the probability mass function f assigns a probability from 0 to 1 to each of these states. To model our conventional notion of probability, we require that the total probability assigned by f to all possible states should sum up to 1.

Definition A.2 (Event)**.** Given a probability space (S, f), an **event** is simply a subset of S. The probability of an event E, denoted by $\Pr_{(S,f)}[E] = \Pr[E]$, is defined to be $\sum_{x \in E} f(x)$. In particular, the event that includes "everything", $E = S$, has probability $\Pr[S] = 1$.

Even though events and probabilities are not well-defined without a probability space, by convention, we often omit S and f in our statements when they are clear from context.

Example A.3. Consider rolling a regular 6-sided die. The sample space is $S = \{1, 2, 3, 4, 5, 6\}$, and the probability mass function is constant: $f(x) = 1/6$ for all $x \in S$. We refer to such a probability mass function (i.e., one that is constant) as being *equiprobable*. The event of an even roll is $E = \{2, 4, 6\}$, and this occurs with probability

$$\Pr[E] = \sum_{x \in \{2,4,6\}} f(x) = \frac{1}{2}$$

Some Basic Properties Now that probability spaces are defined, we give a few basic properties of probability:

Claim A.4. *If A and B are disjoint events ($A \cap B = \emptyset$) then* $\Pr[A \cup B] = \Pr[A] + \Pr[B]$.

Proof. By definition,

$$\begin{aligned} \Pr[A \cup B] &= \sum_{x \in A \cup B} f(x) \\ &= \sum_{x \in A} f(x) + \sum_{x \in B} f(x) && \text{since } A \text{ and } B \text{ are disjoint} \\ &= \Pr[A] + \Pr[B] \end{aligned}$$

■

Corollary A.5. *For any event E,* $\Pr[\overline{E}] = 1 - \Pr[E]$.

Proof. This follows directly from Claim A.4, $\overline{E} \cup E = S$, and $\overline{E} \cap E = \emptyset$. ■

When events are not disjoint, we instead have the following:

Claim A.6. *Given events A and B,* $\Pr[A \cup B] = \Pr[A] + \Pr[B] - \Pr[A \cap B]$.

Proof. First observe that $A \cup B = (A - B) \cup (B - A) \cup (A \cap B)$ and that all the terms on the RHS are disjoint. Therefore

$$\Pr[A \cup B] = \Pr[A - B] + \Pr[B - A] + \Pr[A \cap B] \tag{A.1}$$

Similarly, we have

$$\Pr[A] = \Pr[A - B] + \Pr[A \cap B] \tag{A.2}$$
$$\Pr[B] = \Pr[B - A] + \Pr[A \cap B] \tag{A.3}$$

because, say A is the disjoint union of $A - B$ and $A \cap B$. Substituting (A.2) and (A.3) into (A.1) gives

$$\begin{aligned}\Pr[A \cup B] &= \Pr[A - B] + \Pr[B - A] + \Pr[A \cap B] \\ &= (\Pr[A] - \Pr[A \cap B]) + (\Pr[B] - \Pr[A \cap B]) + \Pr[A \cap B] \\ &= \Pr[A] + \Pr[B] - \Pr[A \cap B]\end{aligned}$$

■

A useful corollary of Claim A.6 is the *Union Bound*.

Corollary A.7 (Union Bound)**.** *Given events A and B,* $\Pr[A \cup B] \leq \Pr[A] + \Pr[B]$. *In general, given events $A_1 \ldots, A_n$,*

$$\Pr\left[\bigcup_i A_i\right] \leq \sum_i \Pr[A_i]$$

A.2 Conditional Probability

Suppose that after receiving a random 5-card hand dealt from a standard 52-card deck, we are told that the hand contains "at least a pair" (that is, at least two of the cards have the same rank). How do we calculate the probability of a full-house given this extra information? Consider the following thought process:

- Start with the original probability space of containing all 5-card hands, pair or no pair.

- To take advantage of our new information, eliminate all hands that do not contain a pair.
- Re-normalize the probability among the remaining hands (that contain at least a pair).

Motivated by this line of reasoning, we define conditional probability as follows:

Definition A.8. Let A and B be events, and let $\Pr[B] \neq 0$. The conditional probability of A, conditioned on B, denoted by $\Pr[A \mid B]$, is defined as

$$\Pr[A \mid B] = \frac{\Pr[A \cap B]}{\Pr[B]}$$

Independence By defining conditional probability, we modeled how the occurrence of one event can affect the probability of another event. An equally important concept is *independence*, where a set of events *do not* affect each other.

Definition A.9 (Independence)**.** A sequence of events $A_1, \ldots, A_n$ are (mutually) independent if and only if for every subset of these events, $A_{i_1}, \ldots, A_{i_k}$,

$$\Pr[A_{i_1} \cap A_{i_2} \cap \cdots \cap A_{i_k}] = \Pr[A_{i_1}] \cdot \Pr[A_{i_2}] \cdots \Pr[A_{i_k}]$$

If there are just two events, A and B, then they are independent if and only if $\Pr[A \cap B] = \Pr[A] \Pr[B]$. The following claim gives justification to the definition of independence.

Claim A.10. *If A and B are independent events and $\Pr[B] \neq 0$, then $\Pr[A \mid B] = \Pr[A]$. In other words, conditioning on B does not change the probability of A.*

Proof.

$$\Pr[A \mid B] = \frac{\Pr[A \cap B]}{\Pr[B]} = \frac{\Pr[A] \Pr[B]}{\Pr[B]} = \Pr[A]$$

■

A.3 Bayes' Rule

Suppose that we have a test against a rare disease that affects only 0.3% of the population, and that the test is 99% effective (i.e., if a person has the disease the test says YES with probability 0.99, and otherwise it says NO with probability 0.99). If a random person in the populous tested positive, what is the probability that he has the disease? The answer is not 0.99. Indeed, this is an exercise in conditional probability: what are the chances that a random person has the rare disease, given the occurrence of the event that he tested positive?

We start with with some preliminaries.

Claim A.11. *Let $A_1, \ldots, A_n$ be disjoint events with non-zero probability such that $\bigcup_i A_i = S$ (i.e., the events are exhaustive; the events partition the sample space S). Let B be an event. Then* $\Pr[B] = \sum_{i=1}^{n} \Pr[B \mid A_i] Pr[A_i]$

Proof. By definition $\Pr[B \mid A_i] = \Pr[B \cap A_i] / \Pr[A_i]$, and so the RHS evaluates to

$$\sum_{i=1}^{n} \Pr[B \cap A_i]$$

Since $A_1, \ldots, A_n$ are disjoint it follows that the events $B \cap A_1, \ldots, B \cap A_n$ are also disjoint. Therefore

$$\sum_{i=1}^{n} \Pr[B \cap A_i] = \Pr\left[\bigcup_{i=1}^{n} B \cap A_i\right] = \Pr\left[B \cap \bigcup_{i=1}^{n} A_i\right] = \Pr\left[B \cap S\right] = \Pr[B]$$

■

Theorem A.12 (Bayes' Rule)**.** *Let A and B be events with non-zero probability. Then:*

$$\Pr[B \mid A] = \frac{\Pr[A \mid B] \Pr[B]}{\Pr[A]}$$

Proof. Multiply both sides by $\Pr[A]$. Now by definition of conditional prob, both sides equal:

$$\Pr[B \mid A] \Pr[A] = \Pr[A \cap B] = \Pr[A \mid B] \Pr[B]$$

■

Sometimes, it is useful to expand the statement of Bayes' Rule with Claim A.11:

Corollary A.13 (Bayes' Rule Expanded)**.** *Let A and B be events with non-zero probability. Then:*

$$\Pr[B \mid A] = \frac{\Pr[A \mid B]\Pr[B]}{\Pr[B]\Pr[A \mid B] + \Pr[\overline{B}]\Pr[A \mid \overline{B}]}$$

Proof. We apply Claim A.11, using that B and $\overline{B}$ are disjoint and $B \cup \overline{B} = S$. ■

We return to our original question of testing for rare diseases. Let us consider the sample space $S = \{(t, d) \mid t \in \{0, 1\}, d \in \{0, 1\}\}$, where t represents the outcome of the test on a random person in the populous, and d represents whether the same person carries the disease or not. Let D be event that a randomly drawn person has the disease ($d = 1$), and T be the event that a randomly drawn person tests positive ($t = 1$).

We know that $\Pr[D] = 0.003$ (because 0.3% of the population has the disease). We also know that $\Pr[T \mid D] = 0.99$ and $\Pr[T \mid \overline{D}] = 0.01$ (because the test is 99% effective). Using Bayes' rule, we can now calculate the probability that a random person, who tested positive, actually has the disease:

$$\begin{aligned}\Pr[D \mid T] &= \frac{\Pr[T \mid D]\Pr[D]}{(\Pr[D]\Pr[T \mid D] + \Pr[\overline{D}]\Pr[T \mid \overline{D}])} \\ &= \frac{.99 * .003}{.003 * .99 + .997 * .01} = 0.23\end{aligned}$$

Notice that 23%, while significant, is a far cry from 99% (the effectiveness of the test). This final probability can vary if we have a different *prior* (initial belief). For example, if a random patient has other medical conditions that raises the probability of contracting the disease up to 10%, then the final probability of having the disease, given a positive test, raises to 92%.

Updating Beliefs after Multiple Signals Our treatment so far discusses how to update our beliefs after receiving one signal (the outcome of the test). How should we update if we receive multiple signals? That is, how do we compute $\Pr[A \mid B_1 \cap B_2]$? To answer this question, we first need to define a notion of conditional independence.

Definition A.14 (Conditional Independence)**.** A sequence of events $B_1, \ldots, B_n$ are conditionally independent given event A if and only if for every subset of the sequence of events, $B_{i_1}, \ldots, B_{i_k}$,

$$\Pr\left[\bigcap_k B_{i_k} \mid A\right] = \prod_k \Pr[B_{i_k} \mid A]$$

In other words, given that the event A has occurred, then the events $B_1, \ldots, B_n$ are independent.

When there are only two events, B_1 and B_2, they are conditionally independent given event A if and only if $\Pr[B_1 \cap B_2 \mid A] = \Pr[B_1 \mid A] \Pr[B_2 \mid A]$.

If the signals we receive are conditionally independent, we can still use Bayes' rule to update our beliefs. More precisely, if we assume that the signals B_1 and B_2 are independent when conditioned on A, and also independent when conditioned on $\overline{A}$, then:

$$\begin{aligned} & \Pr[A \mid B_1 \cap B_2] \\ = & \frac{\Pr[B_1 \cap B_2 \mid A] \Pr[A]}{\Pr[A] \Pr[B_1 \cap B_2 \mid A] + \Pr[\overline{A}] \Pr[B_1 \cap B_2 \mid \overline{A}]} \\ = & \frac{\Pr[B_1 \mid A] \Pr[B_2 \mid A] \Pr[A]}{\Pr[A] \Pr[B_1 \mid A] \Pr[B_2 \mid A] + \Pr[\overline{A}] \Pr[B_1 \mid \overline{A}] \Pr[B_2 \mid \overline{A}]} \end{aligned}$$

In general, given signals $B_1, \ldots, B_n$ that are conditionally independent given A and conditionally independent given $\overline{A}$, we have

$$\Pr\left[A \mid \bigcap_i B_i\right] = \frac{\Pr[A] \prod_i \Pr\left[B_i \mid A\right]}{\Pr[A] \prod_i \Pr\left[B_i \mid A\right] + \Pr[\overline{A}] \prod_i \Pr\left[B_i \mid \overline{A}\right]}$$

Spam Detection Using "training data" (e-mails classified as spam or not by hand), we can estimate the probability that a message contains a certain string conditioned on being spam (or not), e.g., Pr["viagra" | spam], Pr["viagra" | not spam]. We can also estimate the probability that a random e-mail is spam, i.e., Pr[spam] (this is about 80% in real life, although most spam detectors are "unbiased" and assume Pr[spam] = 50% to make calculations nicer).

By choosing a diverse set of keywords, say $W_1, \ldots, W_n$, and assuming that the occurrence of these keywords are conditionally independent given a spam message or given a non-spam e-mail, we can use Bayes' rule to estimate the probability that an e-mail is spam based on the words it contains (we have simplified the expression assuming Pr[spam] = Pr[not spam] = 0.5):

$$\Pr\left[\text{spam} \mid \bigcap_i W_i\right] = \frac{\prod_i \Pr\left[W_i \mid \text{spam}\right]}{\prod_i \Pr\left[W_i \mid \text{spam}\right] + \prod_i \Pr\left[W_i \mid \text{not spam}\right]}$$

A.4 Random Variables

We use *events* to express whether a particular class of outcomes has occurred or not. Sometimes we want to express more: for example, after 100 fair coin tosses, we want to study how many coin tosses were heads (instead of focusing on just one event, say, that there were 50 coin tosses). This takes us to the definition of *random variables*.

Definition A.15. A random variable X on a probability space (S, f) is a function from the sample space to the real numbers $X : S \to \mathbb{R}$.

So, in the example of 100 coin tosses, given any outcome of the experiment $s \in S$, we would define $X(s)$ to be the number of heads that occurred in that outcome.

Definition A.16. Given a random variable X on probability space (S, f), we can consider a new probability space (S', f_X) where the sample space is the range of X, $S' = \{X(s) \mid s \in S\}$, and the probability mass function is extended from f, $f_X(x) = \Pr_{S,f}[\{x \mid X(s) = x\}]$. We call f_X the *probability distribution* or the *probability density function* of the random variable X.

Example A.17. Suppose we toss two 6-sided dice. The sample space would be pairs of outcomes, $S = \{(i, j) \mid i, j \in \{1, \ldots, 6\}\}$, and the probability mass function is equiprobable. Consider the random variables, $X_1(i, j) = i$, $X_2(i, j) = j$ and $X(i, j) = i + j$. These random variables denotes the outcome of the first die, the outcome of the second die, and the some of the two die, respectively. The probability density function of X would take values:

$$
\begin{aligned}
f_X(1) &= 0 \\
f_X(2) &= \Pr[(1,1)] = 1/36 \\
f_X(3) &= \Pr[(1,2),(2,1)] = 2/36 \\
&\vdots \\
f_X(6) &= \Pr[(1,5),(2,3),\ldots,(3,1)] = 5/36 \\
f_X(7) &= \Pr[(1,6)..(6,1)] = 6/36 \\
f_X(8) &= \Pr[(2,6),(3,5),\ldots,(6,2)] = 5/36 = f_X(6) \\
&\vdots \\
f_X(12) &= 1/36
\end{aligned}
$$

Notation Regarding Random Variables We can describe events by applying predicates to random variables (e.g., the event that X, the number of heads, is equal to 50). We often use a short-hand notation, in which we treat random variables as if they are real numbers: if X is a random variable, we let e.g., "$X = 50$" denote the event $\{s \in S \mid X(s) = 50\}$. Using this notation, we may define the probability density function of a random variable X as $f_X(x) = \Pr[X = x]$.

In a similar vein, we can define new random variables from existing random variables. In Example A.17, we can write $X = X_1 + X_2$, to mean that for any $s \in S$, $X(s) = X_1(s) + X_2(s)$ (again, the notation treats, X, X_1 and X_2 as if the are real numbers).

Independent Random Variables The intuition behind independent random variables is just like that of events: the value of one random variable should not affect the value of another independent random variable.

Definition A.18. A sequence of random variables $X_1, X_2, \ldots, X_n$ are (mutually) independent if for every subset $X_{i_1}, \ldots, X_{i_k}$ and for any real numbers $x_1, x_2, \ldots, x_k$, the events $X_1 = x_{i_1}, X_2 = x_{i_2}, \ldots, X_{i_k} = x_k$ are (mutually) independent.

In the case of two random variables X and Y, they are independent if and only if for all real values x and y, $\Pr[X = x \cap X = y] = \Pr[X = x] \Pr[Y = y]$.

A common use of independence is to model the outcome of consecutive coin tosses: Consider a biased coin that comes up heads with probability p. Define $X = 1$ if the coin comes up heads and $X = 0$ if the coin comes up tails; then X is called the Bernoulli random variable with probability p. Suppose now we toss this biased coin n times, and let Y be the random variable that denotes the total number of occurrence of heads. We can view Y as a sum of independent random variables, $\sum_{i=1}^{n} X_i$, where X_i is a Bernoulli random variable with probability p that represents the outcome of the i^{th} toss. We leave it as an exercise to show that the random variables $X_1, \ldots, X_n$ are indeed independent.

A.5 Expectation

Given a random variable defined on a probability space, what is its "average" value? Naturally, we need to weigh things according to the probability that the random variable takes on each value.

Definition A.19. Given a random variable X defined over a probability space (S, f), we define the expectation of X to be

$$\mathbb{E}[X] = \sum_{x \in Range(X)} \Pr[X = x] \cdot x = \sum_{x \in Range(X)} f_X(x) \cdot x$$

An alternative but equivalent definition is

$$\mathbb{E}[X] = \sum_{s \in S} f(s) X(s)$$

These definitions are equivalent because:

$$\begin{aligned}
&\sum_{x \in Range(X)} \Pr[X = x] \cdot x \\
&= \sum_{x \in Range(X)} \sum_{s \in S \text{ s.t. } X(s)=x} f(s) \cdot x \\
&= \sum_{x \in Range(X)} \sum_{s \in S \text{ s.t. } X(s)=x} f(s) \cdot X(s) \\
&= \sum_{s \in S} f(s) X(s)
\end{aligned}$$

The following fact can be shown with a similar argument:

Claim A.20. *Given a random variable X and a function $g : \mathbb{R} \to \mathbb{R}$,*

$$\mathbb{E}[g(X)] = \sum_{x \in Range(X)} \Pr[X = x] g(x)$$

Proof.

$$\begin{aligned}
&\sum_{x \in Range(X)} \Pr[X = x] g(x) \\
&= \sum_{x \in Range(X)} \sum_{s \in S \text{ s.t. } X(s)=x} f(s) g(x) \\
&= \sum_{x \in Range(X)} \sum_{s \in S \text{ s.t. } X(s)=x} f(s) g(X(s)) \\
&= \sum_{s \in S} f(s) g(X(s)) = \mathbb{E}[g(X)]
\end{aligned}$$

■

Example A.21. Suppose in a game, with probability 1/10 we are paid \$10, and with probability 9/10 we are paid \$2. What is our expected payment? The answer is

$$\frac{1}{10}\$10 + \frac{9}{10}\$2 = \$2.80$$

An Application to Decision Theory In decision theory, we assign a real number, called the *utility*, to each outcome in the sample space of a probabilistic game. We then assume that *rational* players make decisions that maximize their *expected utility*. For example, should we be willing to pay \$2 to participate in the game in Example A.21? If we assume that our utility is exactly the amount of money that we earn, then

with probability 1/10 we get paid \$10 and gets utility 8
with probability 9/10 we get paid \$2 and gets utility 0

This gives a positive expected utility of 0.8, so we should play the game!

This reasoning of utility does not always explain human behavior though. Suppose there is a game that cost a thousand dollars to play. With one chance in a million, the reward is two billion dollars (!), but otherwise there is no reward. The expected utility is

$$\frac{1}{10^6}(2 \times 10^9 - 1000) + (1 - \frac{1}{10^6})(0 - 1000) = 1000$$

One expects to earn a thousand dollars from the game on average. Would you play it? Turns out many people are *risk-averse* and would turn down the game. After all, *except with one chance in a million*, you simply lose a thousand dollars. This example shows how expectation does not capture all the important features of a random variable, such as how likely the random variable is to end up close to its expectation (in this case, the utility is either -1000 or two billion, not close to the expectation of 1000 at all).

In other instances, people are risk-seeking. Take yet another game that takes a dollar to play. This time, with one chance in a billion, the reward is a million dollars; otherwise there is no reward. The expected utility is

$$\frac{1}{10^9}(10^6 - 1) + (1 - \frac{1}{10^9})(0 - 1) = -0.999$$

Essentially, to play the game is to throw a dollar away. Would you play the game? Turns out many people do; this is called a lottery!

How can we justify these behaviors in the expected utility framework? The point is that utility may not always be linear in money received/spent. *Non-linear utility functions* may be used to reconcile observed behavior with expected utility theory (but doing so is outside the scope of this course).

Linearity of Expectation One nice property of expectation is that the expectation of the sum of random variables, is the sum of the expectations. This can often simplify the calculation of expectations.

Theorem A.22. *Let $X_1, \ldots, X_n$ be random variables, and $a_1, \ldots, a_n$ be real constants. Then*

$$\mathbb{E}\left[\sum_{i=1}^{n} a_i X_i\right] = \sum_{i=1}^{n} a_i \, \mathbb{E}[X_i]$$

Proof.

$$\begin{aligned}
\mathbb{E}\left[\sum_{i=1}^{n} a_i X_i\right] &= \sum_{s \in S} f(s) \sum_{i=1}^{n} a_i X_i(s) \\
&= \sum_{s \in S} \sum_{i=1}^{n} a_i f(s) X_i(s) \\
&= \sum_{i=1}^{n} a_i \sum_{s \in S} f(s) X_i(s) \\
&= \sum_{i=1}^{n} a_i \, \mathbb{E}[X_i]
\end{aligned}$$

■

Example A.23. If we make n tosses of a biased coin that ends up heads with probability p, what is the expected number of heads? Let $X_i = 1$ if the i^{th} toss is heads, and $X_i = 0$ otherwise. Then, X_i is an independent Bernoulli random variable with probability p, and has expectation

$$\mathbb{E}[X_i] = p \cdot 1 + (1 - p) \cdot 0 = p$$

The expected number of heads is then

$$\mathbb{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathbb{E}[X_i] = np$$

Thus if the coin was fair, we would expect $(1/2)n$, half of the tosses, to be heads.

Conditional Expectations We may also define a notion of an expectation of a random variable X conditionned on some event H by simply conditioning the probability space on H:

Definition A.24. Given a random variable X defined over a probability space (S, f), and some event H on S, we define the expectation of X conditionned on H to be

$$\mathbb{E}[X \mid H] = \sum_{x \in H} \Pr[X = x \mid H] \cdot x$$

We end this section by showing an analog of A.11 for the case of expectations.

Claim A.25. *Let $A_1, \ldots, A_n$ be disjoint events with non-zero probability such that $\bigcup_i A_i = S$. Let X be a random variable over S. Then*

$$\mathbb{E}[X \mid S] = \sum_{i=1}^{n} \mathbb{E}[X \mid A_i] Pr[A_i \mid S]$$

Proof. By definition $\mathbb{E}[X \mid A_i] = \sum_x \Pr[X = x \mid A_i] \cdot x$ and so the RHS evaluates to

$$\begin{aligned}
&\sum_{i=1}^{n} \sum_{x} \Pr[X = x \mid A_i] \Pr[A_i \mid S] \cdot x = \\
&= \sum_{i=1}^{n} \sum_{x} \frac{\Pr[X = x \cap A_i]}{\Pr[A_i]} \frac{\Pr[A_i \cap S]}{\Pr[S]} \cdot x = \\
&= \sum_{i=1}^{n} \sum_{x} \frac{\Pr[X = x \cap A_i]}{\Pr[S]} \cdot x = \\
&= \sum_{x} \frac{\Pr[X = x \cap S]}{\Pr[S]} \cdot x = \\
&= \sum_{x} \Pr[X = x \mid S] \cdot x
\end{aligned}$$

which equals the LHS. ■