A Course in Networks and Markets

Rafael Pass
TA: Andrew Morgan

Last updated: November 30, 2016

Introduction

In this course, we will explore the connections between the worlds of computer science, economics, and sociology, using tools from game theory and graph theory. We focus how network structure and network effects play a role in economic markets. In this section we provide an overview of the course through a few examples.

Example 1: Markets with Network Effects—iPhone vs. Android Consider a market with two competing mobile phones. Each of the phones has some *intrinsic* value to a buyer, but the *actual* value of each model to the buyer varies according to how many of their friends have the same phone—this is referred to a *network effect*): For instance, if a large number of my friends have an iPhone, but I have an Android phone, I might be inclined to switch to the iPhone myself.

Some questions that might arise:

- Will there eventually be a stable solution, where everyone is happy with the phone they have?
- What will this solution look like? (Will everyone eventually have the same phone, or can we get a market for both?)
- If I want to market iPhones, to which "influential" individuals should I offer discounts in order to most efficiently take over the market?

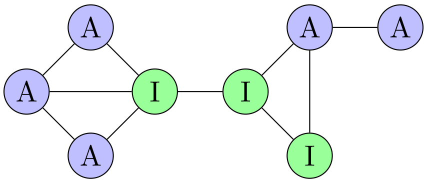


Figure 0.1: A very small Android/iPhone network example.

- How should I set the price of my phone to best market it? (Perhaps start low and increase the price as more people buy it due to the network effect?)

We will study models that allow us to answer these questions.

Example 2: The Role of Beliefs. In fact, in the example above, it may suffice for marketing a phone that enough people to simply *believe* that their friends will want the phone. If people believe that their friends will buy the phone (commonly accomplished by advertising for products with such a network effect), their perceived value of the phone will dramatically increase, and they will be far 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 a phone—that is, so-called *higher-level beliefs* can have a huge effect. We will study models for discussing and analyzing such higher-level beliefs and their effectiveness. Due to the above reasons, markets with large network effect are quite vulnerable to valuation *bubbles* and *crashes*; we will investigate how these occur.

More generally, we will discuss how crowds can 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 (for instance, prediction teams outperforming single analysts in elections).

- *The foolishness of crowds.* In other situations, "misinformation" can be circulated through networks in "information cascades" (e.g. urban legends being popularized through social media).

Example 3: Matching Markets, Auctions and Voting. Let's say we have three people and three houses called A, B, C. Let's also assume that everyone likes house A better than house B and houses B, C better than house C (i.e., $A > B > C$). How should we match people to houses? The key here is to set prices for the three houses according to how much the three people value them; if we don't do this, two people will be upset (or homeless!) no matter how we match people with houses. So,

- can we set prices for these three houses so that everyone will be happy with the outcome? (Actually, *yes*; we will show that such "market-clearing prices" are guaranteed to exist.)

- can we set prices so that the three people want to *truthfully* report how much each house is worth to them when bidding? (Also *yes*; the famous Vickrey-Clark-Groves auction mechanism provides a way to do this.)

In fact, these methods 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 shown)—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. 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. Of course, there are plenty of ways to manipulate this scheme (and indeed, this is why companies attempt so-called "search engine optimization", to increase their visibility with respect to the search algorithm). We will discuss this, and more generally, why voting schemes (used e.g., in presidential elections) are susceptible to "strategic" voting where voters often are incentivized to not report their truthful preferences (for instance, what if your favorite candidate in the US presidential election is a third-party candidate?).

Outline of the course. We will start by providing brief introductions to the following mathematical theories:

- **Game Theory.** The study of how strategic and rational agents (trying to maximize their utility) interact.

- **Graph Theory.** The study of *graphs*, mathematical constructs used to model networks of interconnected nodes.

We then use tools from them to provide answers to the above questions.

Contents

Chapter 1

Game Theory

In this chapter, we will develop tools for reasoning about how strategic agents act in order to maximize their utility in a predefined situation. We begin with one of the most basic and well-known problems in the field.

1.1 The Prisoner's Dilemma

In this one of many versions of the famous thought-experiment, two robbers robbed a bank and managed to escape and hide their ill-gotten gains, but were caught afterwards. They are put in separate rooms and given the choice to either *cooperate* (stay silent) or *defect* (accuse their partner).

For each player, and each outcome of the game—that is, the choices (or *actions*) of both player—we define the *utility*, or favorability, of outcome:

- If both robbers *defect*, they are both charged as guilty and given 4 years in prison; we write $u_1(D, D) = u_2(D, D) = -4$, or simply $u(D, D) = (-4, -4)$.
- If both *cooperate* and remain silent, there is no evidence to charge them, and they go free and share the money they stole. We write $u(C, C) = (0.5, 0.5)$
- However, if one robber cooperates and the other defects and accuses him, that robber receives 10 years in prison, while the other goes free and receives all of the money. We write $u(C, D) = (-10, 1)$ for the scenario where the first stays silent and is sold out by the second, and $u(D, C) = (1, -10)$ for the inverse.

We can write the outcome space for this game as a grid, where the first player's choices are represented on the right side and the second player's choices on the top side:

	$(*, C)$	$(*, D)$
$(C, *)$	(0.5, 0.5)	(-10, 1)
$(D, *)$	(1, -10)	(-4, -4)

Notice that, no matter what the other robber decides to do, each robber gets more utility from defecting rather than cooperating! ($u_1(D, *) > u_1(C, *)$ and $u_2(*, D) > u_2(*, C)$ regardless of what action the $*$ represents). Thus, one would expect both robbers to defect.

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

1.2 Normal-form games

We will focus on a small subset of games, where the following properties hold:

- Players move 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.

Definition 1.1. A (finite) **normal-form game** is a tuple $G = (n, A, u)$, 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 *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*. We use the term *action* and *strategy* interchangeably. Some more notational details:

- Let $[n] = [1, 2, \dots, n]$;
- 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:
  - $u(C, C) = (0.5, 0.5)$ (i.e., $u_1(C, C) = u_2(C, C) = 0.5$);
  - $u(C, D) = (-10, 1)$;
  - $u(D, C) = (1, -10)$;
  - $u(D, D) = (-4, -4)$.

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.

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 have managed to put the criminals (assuming they act rationally) in prison! But consider a similar game where, for instance, two countries are deciding whether to cooperate or fight one another (defect) with 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 $(0.5, 0.5)$. (As a side note, let us remark that when people actually play this game, not everyone defects. In fact, the rate of cooperation/defection varies by culture. There are models to deal with this; see e.g., Halpern-Pass: "Games with Translucent Players".)

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:

	$(*, L)$	$(*, M)$	$(*, R)$
$(U, *)$	(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 game. Since M is terrible "no matter what", we should remove it from consideration. Let's thus consider the game resulting from removing this action.

	$(*, L)$	$(*, R)$
$(U, *)$	(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:

	$(*, L)$	$(*, R)$
$(U, *)$	(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, initialize their "initial" action space $A_i^0 = A$.
- 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}. (That is, A_i^j is obtained by taking A_i^{j-1} and removing all strategies 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 procedue always ends with a non-empty set.

ISD and Common Knowledge of Rationality Intuitively, no "rational" players should every 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. If everyone *knows* that everyone is rational, we know that nobody plays a strictly dominated strategy (i.e., restrict the game by removing all strictly dominated strategies), and then we remove all strictly dominated strategies in the restricted game. Thus, intuitively, if we have *common knowledge of rationality*—that is, everyone knows that everyone knows . . . that everyone is rational, people can only plays strategies that survive ISD. Later on in the course, 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 or 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:

Coordination Game: Bach-Stravinsky (a.k.a. Battle of the Sexes) 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 don't go together. Formally, we can say $u(B, B) = (2, 1)$, $u(S, S) = (1, 2)$, 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, 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 utility, whereas player 2 would also lose 1 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 $a'_i \in A_i$,

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

So, assuming that everyone else sticks to the equilibrium strategy, nobody can increase their own utility by deviating from that. In other words, there does not exists some player i that has a "profitable deviation".

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 $\vec{a}$. Assume for the sake of contradiction that there is a player i such that a_i is eliminated; consider the round j where this happens for the first time. Then $a_{-i} \in A_{-i}^{j-1}$ (since at that point a_i had not been 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 a is a PNE. ■

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

Proof. Consider some game where a unique strategy profile $\vec{a}$ survives. Assume for the sake of contradiction that a is *not* a PNE. Then there must 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, it must have been deleted at some round j—that is, there must exists some strategy a''_i which dominates a'_i w.r.t. some set A_{-i}^j which contains a_{-i} (since a_{-i} survives til the end). But then (by transitivity of strict dominance) a''_i also strictly dominates a_i; thus a_i would also have been deleted in the same round, which is a contradiction. ■

Best responses Another way to think of PNEs is in terms of a *best response* to a strategy. Given an action profile $\vec{a}$, let $B_i(\vec{a})$— i's best-response set—be the set of strategies in $a'_i \in A_i$ that maximize player i's utility given a_{-i} (that is, that maximize $u(a'_i, a_{-i})$). Let $B(\vec{a}) = \times_{i\in[n]} B_i(\vec{a})$ (i.e., $\vec{a} \in B(\vec{a})$ if and only if for all $i \in [n]$, $a_i \in B_i(\vec{a})$.)

The following claim is almost immediate from the definition.

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

Proof. This follows immediately from the fact that the existence of a "profitable deviation" from a strategy profile $\vec{a}$ is equivalent to the existence of a player i and some action a'_i such that $a'_i \notin B(\vec{a})$. ■

But 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 B. (Looking forward, the notion of a fixed-point will be extremely 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 a stable outcome of play by players that "know" each other and how the other player will play in the game. 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 way to learn is to start at some arbitrary action profile 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 else does exactly what they previously did, and best respond to this. We refer to this process as *best-response dynamics (BRD)* and formalize it as follows:

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

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.

While BRD works for many games of interest, the procedure does not always converge. In fact, there are some games that do not even have a PNE—for instance, think of rock-paper-scissors. 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 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 will converge to the strictly dominant action profile.

Claim 1.10. *Consider an n-player game 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 dominant strategy and will never ever switch again. Thus after at most n round everyone has switched to their strictly dominant action a_i. ∎

Mixed-strategy Nash Equilibrium As mentioned, not all game have 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 *probability distribution* over actions (i.e. pick a "weight" with which to randomly choose each possible action). John Nash's celebrated theorem shows that every game with a finite action space and finite number of players have a mixed-strategy NE (even if it may not have a PNE).

Let us mention that "perfectly" randomizing across a mixed-strategy distribution may now always be easy. (For instance, think about picking each of rock, paper, and scissors with probability $\frac{1}{3}$, the mixed strategy Nash equilibrium for that game; if it were trivial to truly randomize, there would not be such things as extremely skilled players and world championships for the game!) In fact, 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; see [Halpern-Pass: "Game theory with costly computation"].

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).

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 the effect of BRD.

A cautionary game: The Traveler's Dilemma Two travelers fly to China and buy identical vases while there. However, on the flight back, both vases are broken; the airline company wants to reimburse them, but doesn't know how much the vase cost. So they put each of the travelers in separate rooms and ask them to say how much their vase cost (from \$2 to \$100).

- If both travelers say the same price, they both get that amount.
- If 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 outcome $(97, 98), (97, 96), (95, 96)...(3, 4), (3, 2), (2, 2)$, 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? In experimental results, 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 punishment/reward as "too small" to start "undercutting"; indeed, other experiments have shown that if we increase the punishment/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.

Chapter 2

Graphs

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 (for instance, what is the fastest route between two computers?).

- Consider a graph of a map, where the edges are roads and the nodes are points of intersection and cities. How could you find the shortest path from point A to point B? (Some roads are one-way, and we 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 applications to social networks. We return to the other applications later on in the course.

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

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

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 $(v, v') \in E$.

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., starts and ends at the same vertex). The length of the walk, path or cycle is k (i.e., the number of edges).

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

2.2 Vertex Degree

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 coming into it (i.e., of the form $(u, v), u \in V$); the **out-degree** is the number of edges going out of it (i.e., of the form $(v, u), u \in V$). The **degree** of v is the sum of the in-degree and the out-degree.

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

We denote the degree of vertex $v \in V$ by $\deg(v)$.

This seemingly cumbersome definition actually makes a lot of sense pictorially: the degree of a vertex corresponds to the number of "lines" connected to 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 or 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 2.5. *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 $2|E| = \sum_{v \in V} \deg(v)$. ■

A useful corollary is the "handshake lemma":[1]

Corollary 2.6. *In a graph, 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 2.5,

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

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. ■

[1] The name stems from the anecdote that the number of people that shake hands with an odd number of people is even.

2.3 Connectivity

We turn to consider the notion of *connectivity.*

Definition 2.7. An undirected graph 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)).

The notion of connectivity on a directed graph is more complicated, because paths are not reversible.

Definition 2.8. A directed graph $G = (V, E)$ is **strongly connected** if there exists a path from any node u to any node v. It is called **weakly connected** if there exists a path from an node u to any node v in the underlying undirected graph: the graph $G' = (V, E')$ where each edge $(u, v) \in E$ in G induces an undirected edge in G' (i.e. $(u, v), (v, u) \in E'$).

When a graph is not connected (or strongly connected), we can decompose the graph into smaller connected components.

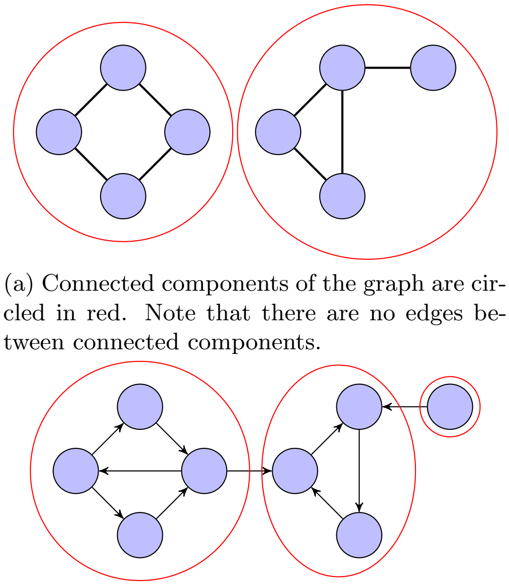
Definition 2.9. Given a graph $G = (V, E)$, a **subgraph** of G is simply a graph $G' = (V', E')$ with $V' \subseteq V$ and $E' \subseteq (V' \times V') \cap E$; we denote subgraphs using $G' \subseteq G$.

A **connected component** of graph $G = (V, E)$ is a *maximal* connected subgraph. i.e., it 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.

We may similarly define a **strongly connected component** as a *maximal* strongly connected subgraph.

Here are two interesting applications of connectivity:

- In social networks, not everyone is in the same connected component. But most people in fact are; we informally refer to this component as the *giant component.* One reason for why this is the case is that if we had two (or more) large components, then all it takes to merge them into a single larger component is for one of the people in each component to become friends—it is very unlikely that this doesn't happen for any such pair of people.
- Paths and connectivity provide a interesting way to visualize the Internet as a "bow-tie" shape. The central portion of this interesting object represents a single giant strongly-connected component (in a directed graph where edges represent links between pages). There is a large cluster of



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

(b) *Strongly* connected components of the graph are circled in red. Note that there can still be edges between strongly connected components.

Figure 2.2: Example of connected components.

nodes directed into the component, a large cluster of nodes directed out of the component, directed "tubes" from the "in" to the "out" clusters, undirected "tendrils", and other disconnected components.

2.4 Shortest Paths and Finding Connected Components

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 traverses a graph as follows: starting from a vertex v, the algorithm marks v as visited, and traverses the neighbors of v (nodes that share an edge with v). After vising the neighbors of v, the algorithm recursively visits the neighbors of

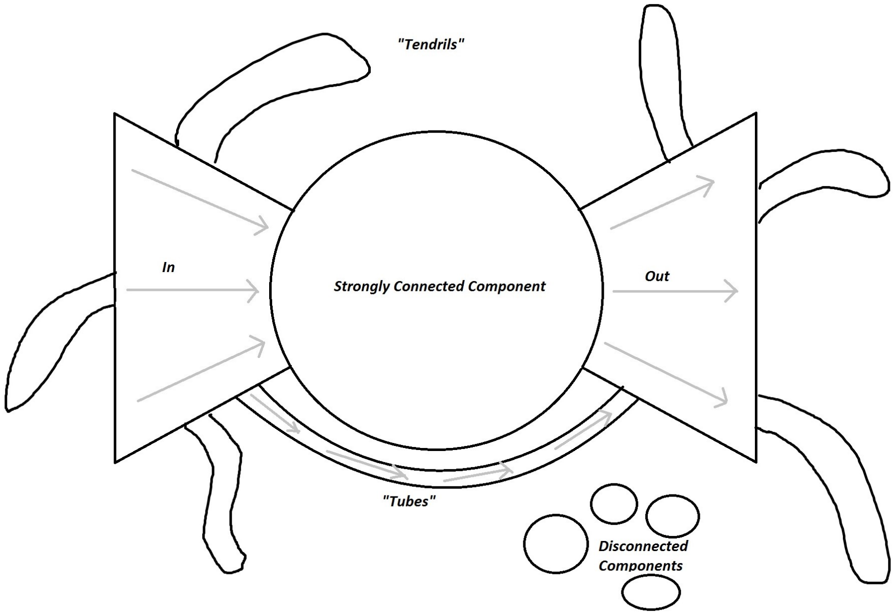


Figure 2.3: An “artist”’s rendition of the graph-theoretic visualization of the Internet. Image from fig. 13.7/source [80] of Easley-Kleinberg.

the neighbors, but takes care to ignore nodes that have been visited before. (There is an alternative method of searching a graph called *depth-first search*, which first fully explores one path and then backtracks to discover others.)

Claim 2.10. *BFS eventually visits vertex v' if and only if there is a path from v to v'.*

Proof. First note that if there is no path between v and v', then of course the search algorithm will never reach v'. On the other hand, assume that there exists a path between v and v', but for the sake of contradiction, that the BFS algorithm does not visit v' after all the reachable vertices are visited. Let v'' be the first node on the path from v to v' that is not visited by BFS (such a node must exists because v' is not visited). We know $v'' \neq v$ since v is visited right away. Let $\tilde{v''}$ be the vertex before v'' on the path from v to v', which must be visited because v'' is the *first* unvisited vertex on the path. But this gives a contradiction; after BFS visits $\tilde{v''}$, it must also visit v'' since v'' is an unvisited neighbor of $\tilde{v''}$. ■

This argument can be extended to show that in fact, BFS would traverse a *shortest* path from v to v'; we just need to rely on the additional observation that in the shortest path we never pass through the same node twice (or else we could just remove the loop between the first and the second pass):

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

BFS can be used to find the connected components of an undirected graph, as follows: Start at any node and explore from there using a BFS; mark all the visited nodes as component 1. Then start a new BFS from any unvisited node; mark all the nodes the new BFS visits as component 2. Continue in the same manner until all nodes have been visited. Notice that this algorithm doesn't work for directed graphs (since the existence of a path from v to v' in a directed graph does *not* imply the existence of a path from v' to v).

Claim 2.12. *The connected components of a graph G can be found in time polynomial in $|G|$.*

Proof. In a graph with n nodes and m vertices, breadth-first search must in the worst case explore each node and each vertex, providing $O(m + n)$ runtime. Even though multiple searches may need to be conducted if a graph has multiple components, it will still only be necessary to explore every node and edge exactly once to find each connected component. ■

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

- The "Bacon number" of an actor or actress is the shortest path from the actor or actress to Kevin Bacon on 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 a mathematician to Paul Erdös on the co-authorship graph.
- Milgram's "small-world" experiment (six degrees of separation): everyone is approximately 6 steps away from anyone else on the graph where edges represent personal acquaintances. This is referred to as the "small-world" phenomenon; it was first demonstrated in the famous experiment wherein 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! Strogatz and Watts (from Cornell) demonstrated, in 1998, a mathematical model that accounts for this; we will return to it later.

- Milgram's experiment shows that not only are people connected through short paths, but they also manage to route messages efficiently given only their local knowledge of the graph. Jon Kleinberg (from Cornell's CS department) refined the model of Strogatz and Watts to demonstrate how such routing could be done.

A simplified small-world model. To understand how such "small-world" effects occur, 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 2 is $\left(1-\frac{1}{4}\right)^{n-2} = \left(\frac{3}{4}\right)^{n-2}$. And as there is even a chance that there is a path (v, v') of length 1, the probability that there is no path of length at most 2 is strictly bounded above by $\left(\frac{3}{4}\right)^{n-2}$, which is very small indeed (especially for large n).

What if we look at all pairs of nodes? To do this in a relatively easy manner, let us recall the *Union Bound* of probability theory: A basic fact of probability is that given two events A and B, $\Pr[A \cup B] = Pr[A] + Pr[B] - Pr[A \cap B]$. In particular, this implies that $\Pr[A \cup B] \leq Pr[A] + Pr[B]$; inductively applying this to i events $A_1, \ldots, A_i$ gives what is known as the Union bound:

$$Pr[\cup_i A_i] \leq \Sigma_i Pr[A_i]$$

Now, lets apply this result to the problem at hand. By the Union Bound we have that the probability that there exists *some* pair of nodes that is more than distance 2 apart is

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

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.

2.5 Bridges and Triadic Closure

Bridges and Local Bridges. We are interested in studying how information in a network is propagated, and what nodes are "powerful" with respect to such information flow; we here restrict our attention to undirected graphs. To that end, we define the notion of a *bridge*.

Definition 2.13. Given a graph $G = (V, E)$, we say that an edge $(a, b) \in E$ between two nodes a, b is a **bridge** if a, b are in the same connected component when the edge (a, b) is present but in separate connected components when removing the edge.

Intuitively, if nodes a and b are on a bridge, they are both "powerful" with respect to information propagation—without them information cannot flow between the components A and B.

We may also consider a slightly weaker form of a bridge, referred to as a *local bridge*:

Definition 2.14. We say that an edge (a, b) is a **local bridge** if there does not exist an node c and edges (a, c) and (b, c) (i.e. there is no "triangle" of edges containing both a and b).

Another way of saying this is that a and b do not have any (other) neighbors in common. Individuals that form local bridges are "powerful" in the sense that they are the ones connecting their friends. (Looking forward, we shall see how this informational power also translates to monetary power in bargaining situations: intuitively, nodes forming local bridges have social capital.)

Triadic Closure, and Weak and Strong Ties. An interesting observation made through experiments by Granovetter reveals that such "local bridges" typically are made up of "weak" friendships (i.e., acquaintances). The experiments, interestingly enough, show that these acquaintances are typically much more useful in searching for a job than "real" (closer) friends. Let's examine a simple property of networks that may explain this:

- Consider a social network graph, and label every edge in the network as either a *weak* tie (an acquaintance) or a *strong* tie (friendship).

- Let's say that a node a violates *Strong Triadic Closure* (STC) if for any two nodes b, c there exist strong ties $(a, b), (a, c)$ but no edge (b, c).

Claim 2.15. *If a node a satisfies (i.e., does not violate) STC and a has at least two strong ties, then any local bridge to another node b that a is involved in must be a weak tie.*

Proof. Assume for contradiction that the statement is false. That is, there exists a node b where a local bridge (a, b) in fact is a strong tie. Then, since by assumption there is at least one more strong tie (a, c), STC requires that (b, c) be either a weak or a strong tie, and thus (a, b) is part of a triangle and cannot be a local bridge, which is a contradiction. ■

The Clustering Coefficient. We note that weaker versions of STC can suffice to provide more generalized versions of this claim. More generally, we may informally consider a generalized notion of *triadic closure*, which states that, if there exists an edge between a and b and one between b and c, then it is "likely" that there exists an edge between b and c. The so-called *clustering coefficient* of a node measures exactly how "likely" it is (i.e., measures the "strength" of the triadic closure):

Definition 2.16. The **clustering coefficient** of a node a is the probability that two random neighbors of a are connected (or equivalently, the fraction of the pairs of neighbors that are connected).

A typical measure of triadic closure in a social network is the *average clustering coefficient* (over the nodes in the network).

Evolution of triadic closure. An important experimentally observed phenomenon in social networks is that two nodes (a, b) are more likely to *become* connected the more neighbors (i.e., friends) they have in common. Intuitively, if we have many friends in common, we are more likely to meet each other and form a friendship. Thus triadic closure is more likely to hold for an edge (a, b) if a and b have many friends in common.

Neighborhood Overlap and the strength of ties. Rather than considering the actual *number* of neighbors two nodes a, b have in common, we may also consider the *fraction* of these neighbors they have in common; this leads to the following definition.

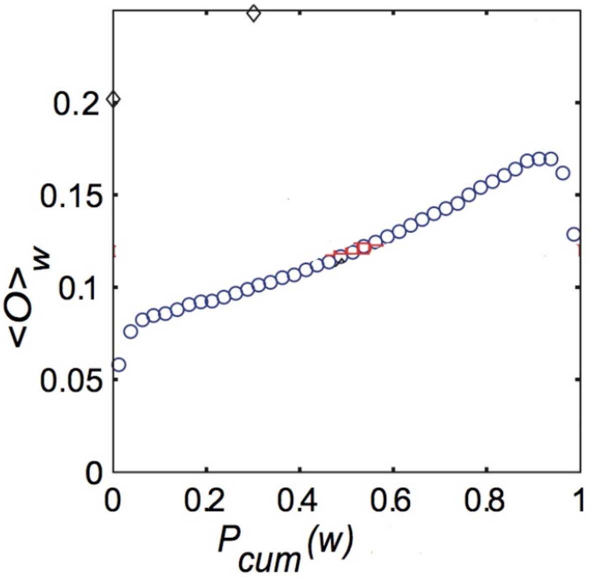


Figure 2.4: A plot of the neighborhood overlap of edges as a function of their percentile with respect to tie strength in a study done using cellphone data. Image from fig. 3.7/source [334] of Easley-Kleinberg.

Definition 2.17. The **neighborhood overlap** of edges connecting a and b is given by the ratio of the number of neighbors of *both* a and b to the number of neighbors of *either* a or b.

Note that a local bridge (a, b) has neighborhood overlap 0 by definition (there can be no neighbors of both a and b). The notion of neighborhood overlap thus provides a generalization of local bridge; such a generalization is useful since in some data sets (e.g. cellphone data), local bridges are rare. Intriguingly, a generalization of Granovetter's observations has been observed with respect to neighborhood overlap: in cellphone data, it appears to grow linearly with the strength of the tie (measured in this case by the number of minutes spent talking to one another on the cell phone).

Chapter 3

Flow and Matching in Graphs

We now consider some additional properties and concepts in graph theory.

3.1 The Max-Flow Problem

Consider a directed graph $G = (V, E)$ and a function $c : E \to \mathbb{R}$ where $c(E)$, for some edge E, is the *capacity* of that edge. Let us label a node $s \in V$ as the *source*, and another node $t \in V$ as the *sink*.

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, etc.) that we can route from s to t respecting the capacity of each edge. Formally, we wish to assign to each edge $e \in E$ a *flow* $f(e)$ such that the following hold:

- *Conservation of flow.* For every node $v \neq s, t$, $\sum_{u|(u,v)\in E} f(u, v) = \sum_{u|(v,u)\in E} f(v, u)$ (i.e. the sum of the flows over incoming edges is the same as over outgoing edges);
- *Edge capacities.* For every edge $e \in E$, $f(e) \leq c(e)$.

We assume here that each capacity is an integer (though there are networks where that is not the case). We can find the maximum flow in a graph using the **Augmenting Path Algorithm**, as follows:

- 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:

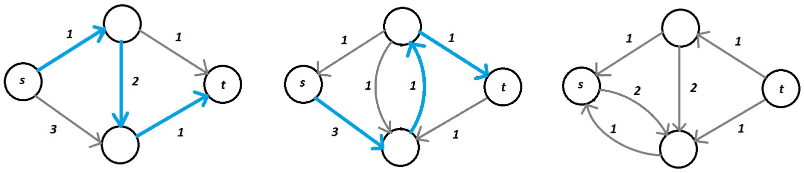


Figure 3.1: Augmenting paths in action. At each stage, we route 1 unit of flow along the augmenting path indicated in blue. Observe that at the end there exist no more augmenting paths, so we have found the maximum flow.

  - 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 3.1 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 3.1 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.

3.2 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 3.1. An (s, t)**-cut** of a flow network graph as defined above is a partition of V into two sets (S, T) such that $s \in S$ and $t \in T$. We define the **capacity** of such a cut to be 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). We refer to a (s, t)-cut as a **min-(s,t)-cut** if its capacity is minimum over all such cuts possible in G.

Theorem 3.2. *Given a graph $G = (V, E)$, integer edge capacities $c : E \to \mathbb{N}$, and two nodes $s, t \in V$, the Augmenting Path Algorithm outputs some maximum (s, t)-flow. Furthemore,*

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

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; thus, if we can prove that the output flow is a maximum flow, we have proven items 1 and 2. We now prove this and along the way 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 is no augmenting path from s to t. Consider the final residual graph G' at this point. Let S denote the set of nodes reachable from s (i.e. the strongly connected component that s belongs to). Let T be the remaining nodes.

- Clearly, $s \in S$; because t is unreachable from s (or else the algortithm 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 a strongly connected component). 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).

- Finaly, 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 value of the max-flow $\leq$ capacity of the min-cut), and in so doing have proven property 3. ■

(We mention that better analyses exist for special instances of the augmenting path algorithm which have an even faster running time for large capacities, but this will not be of importance to us in this chapter.)

3.3 Edge-Disjoint Paths

Let us now observe an elegant 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.

Theorem 3.3. *In any graph G, the number of edge-disjoint paths between any two nodes s,t equals the max-(s,t)-flow in G if all capacities are 1. 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 3.2 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 3.2). 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. ■

3.4 Bipartite Graphs and Maximum Matchings

Let us now see a different application of max-flow to the problem of finding maximum matchings in so-called *bipartite graphs*:

Definition 3.4. 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 $y \in Y$.

In other words, it is a graph where the nodes are divided into two "types" and edges can only exist between a node of one type and a node of the other type. For instance, consider an "affiliation graph" where X is a set of people, Y is a set of universities, and edges (x, y) exist when person x has studied at university y.

Definition 3.5. 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 is found in Figure 3.3. Think of the underlying bipartite graph as specifying the classrooms which are acceptable to the professor, and the matching as an assignment of classrooms to professors.

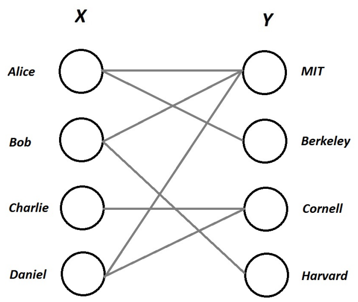


Figure 3.2: A simple example of a bipartite affiliation graph between people and universities.

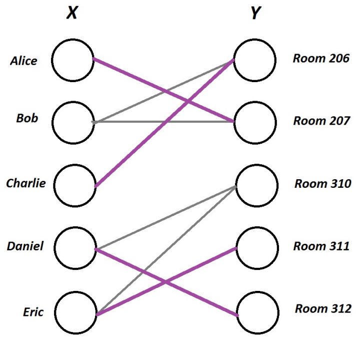


Figure 3.3: A simple example of a maximum matching (of size 4) between professors and classrooms in a bipartite graph.

We now show how to efficiently find maximum matching in bipartite graphs.

Theorem 3.6. *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 *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 3.4 for an illustration.) Set all existing edges from X to Y to have infinite capacity (and, in the residual graph, 0 capacity from Y to

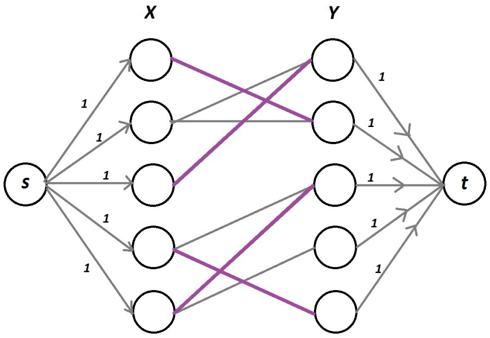


Figure 3.4: The graph from above, expanded to create an s-t flow network. Observe that the maximum flow in this network is equal to the size of the maximum matching.

X). Then find the maximum flow in this graph 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 output by the algorithm induces a matching of size k, and 2) if there exists a matching of size k, then there exists some flow of size k (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 in 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 3.2, 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).

■

3.5 Perfect Matchings and Constricted Sets

Consider now a bipartite graph where $|X| = |Y| = n$. We are interesting in understanding when such a bipartite graph has a *perfect* matching, where every node gets matched.

Definition 3.7. A matching M in a bipartite graph $G = (X \cup Y)$ 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.

For instance, if X is a set of n boys and Y a set of n girls, and the edges (x, y) exist between a boy-girl 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 (a-priori) gives us little insight into *why* a perfect matching is not possible.

The beautiful notion of a *constricted set* turns out to be cruicial for characterizing the existence of perfect matchings. Given a graph $G = (V, E)$, let $N(S)$ denote the set of nodes which are neighbours to *any* nodes 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 S of nodes that has less neigbours than elements in S:

Definition 3.8. Given a bipartite graph $G = (X \cup Y, E)$, a **constricted set** is a subset $S \subseteq X$ such that $|N(S)| < |S|$.

Clearly, if a constricted set exists, no perfect matching can be found—a set of k boys cannot have acceptable marriages with $< k$ girls. Hall's "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.

Theorem 3.9. *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 before, if there exists a constricted set, 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 proves the theorem.

Consider the expanded graph G' from the proof of Theorem 3.6; use the max-flow algorithm from Theorem 3.2 to find the maximum (s, t)-flow, and consider the min-cut (S, T) induced by this flow in Theorem 3.2. Recall that the set S was obtained by considering the strongly connected component in

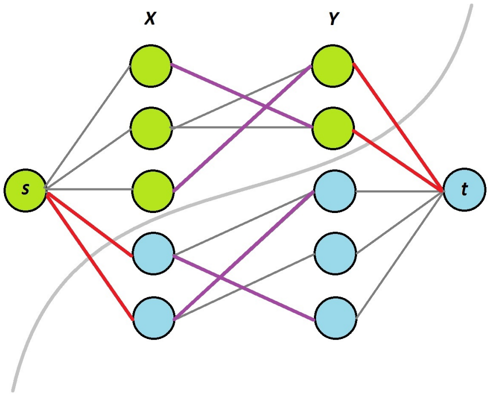


Figure 3.5: Finding a constricted set in the graph from above. Edges in the maximum (not perfect) matching are purple; nodes in S are green, nodes in T are blue, and edges leaving S in the min-cut are red. Observe that $X \cap S$, the set of green nodes on the left side, forms a constricted set.

the final residual graph in the Augmenting Path algorithm; this connected can be found in polynomial time using breadth-first search; see Claim 2.12.

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

- Since G does not have a perfect matching, by Theorem 3.6 (the max-matching, max-flow correspondence) and Theorem 3.2 (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).
- 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 (since those are the only possible edges of capacity 1 that can leave S); see Figure 3.5 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

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

- Combined the above two equations, we have that

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

- 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 $X \cap S$ is a constricted set.

■

Chapter 4

Analyzing Best-Response Dynamics

Recall that a *Pure-Strategy Nash equilibrium (PNE)* is a profile of actions within a game such that no player can improve their utility by deviating from their chosen action. Recall that *best-response dynamics (BRD)* is a process where we attempt to find a Nash equilibrium for a game by selecting an inital action profile $\vec{a}$, iteratively picking players i who are not playing an action a_i in their best-response sets $B_i(\vec{a})$, and having them switch to a best response.

4.1 A Graph Representation of Games

We can use graphs to analyze (normal-form) games as follows: Given a game $G = (n, A, u)$, 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 $\vec{a}$ and $\vec{a'}$ differ only in component i and $a_i \notin B_i(\vec{a})$ but $a'_i \in B_i(\vec{a})$—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 4.1 for an example of such a graph 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 interprereted as simply starting at any node $\vec{u}$ in G' and then picking any outgoing edge, traversing it to reach a new node $\vec{u}'$, picking any outgoing edge from it and so on. That is, we take a walk from any node in the graph.

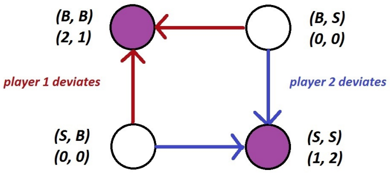


Figure 4.1: A BRD graph for the Coordination Game. “Sinks” (equilibria) are marked in purple. Observe that there are multiple sinks reachable from each non-equilibrium state, depending on which player deviates.

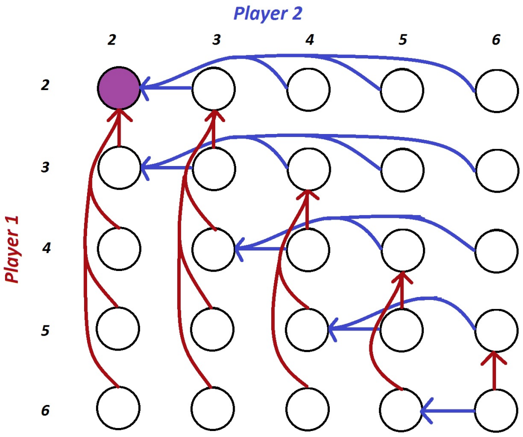


Figure 4.2: A small part of the BRD graph for the Traveler’s Dilemma, illustrating why this game converges to the equilibrium of (2, 2).

- BRD *converges* in 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.)

- $\vec{a}$ is a PNE iff $\vec{a}$ is a *sink* (i.e., a node without any outgoing edges) in G'—nobody can improve its utility by best responding.

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.

4.2 Characterizing Convergence of BRD

We can now use this graph representation to characterize the set of games for which BRD converges. Given a game (n, A, u), we define a **potential** (or "energy") function $\Phi : A \to \mathbb{N}$. A particularly natural potential function (which we will use later) is the *social welfare* (SW) defined as $\Phi(a) = \mathrm{SW}(a) = \Sigma_{i \in n} u_i(a)$—that is the sum of all players' utility.

We can say that a potential function is *ordinal* if, whenever some player wants to deviate from an action profile, the deviation increases the potential.

Definition 4.1. $\Phi : A \to \mathbb{N}$ is an **ordinal potential function** for a game $G = (n, A, u)$ if, for every action profile $a \in A$, every player i, and every action $a'_i \in A_i$, if

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

then

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

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

Theorem 4.2. *BRD converges in G if and only if G has an ordinal potential function.*

Proof. We will prove each direction separately.

Claim 4.3. *If G has an ordinal potential function, then BRD converges for G.*

Proof. Consider an arbitrary starting action profile (node) $\vec{a}$, and consider running BRD starting with a. Each time some player improves their utility, the potential function 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 potential, which guarantees convergence as no more "deviations" are possible from that point. ■

Claim 4.4. *If BRD converges in G, then G has an ordinal potential function.*

Proof. Consider some game G where BRD converges, and consider the induced graph representation G' of G. As noted above, 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 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 (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 $\ell(\vec{a}) \geq k$. 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. ■

Chapter 5

Networked Coordination Games and Contagion

Recall the iPhone/Android game we considered in the introduction to the course. We can use the elements of game theory and graph theory that we have covered thus far to model this game. We begin by considering a simple form of this game on a social network described by an undirected graph.

5.1 Symmetric Networked Coordination Games

Each node on the graph selects an action (a "product"), either A or B, and plays the following *symmetric* (i.e. both players get the same utility in every outcome) coordination game with each of its neighbors (assume $x, y > 0$):

	$(*, A)$	$(*, B)$
$(A, *)$	(x, x)	(0, 0)
$(B, *)$	(0, 0)	(y, y)

Thus:

- if they match, they get the same positive utility $Q(A, A)$ or $Q(B, B)$.
- if they mismatch, they get $Q(A, B) = Q(B, A) = 0$ utility.
- Note that the utility of matching on A may be different from the utility of matching on B. Without loss of generality, we assume $x \geq y$; that is, product A is *objectively no worse* than product B.

The utility of a node is then defined to be the sum of its utilities from all of the games it participates in. Formally,

Definition 5.1. A game $G = (n, A, u)$ is a **symmetric networked coordination game** induced by the graph $G' = (V, E)$ and symmetric coordination utility $Q : \{A, B\} \to R$ if:

- $V = [n] = \{1, 2, \dots, n\}$, $Q(A, B) = Q(B, A) = 0$, $Q(A, A) \geq Q(B, B)$.
- $A = \{A, B\}^n$
- $u_i(\vec{a}) = \sum_{j \in N(i)} Q(a_i, a_j)$, where $N(i)$ is the set of neighbors of i w.r.t. G'.

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

A simple decision rule: The adoption threshold Consider a node v that has d neighbors (friends), a fraction p of whom choose A and the remaining fraction $(1 - p)$ of whom choose B. Then v's utility for choosing A is pdx and its utility for choosing B is $(1 - p)dy$. So, what action should v take to maximize its utility (i.e., to best respond to the actions of the other players)?

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

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

5.2 PNE and BRD in Symmetric Games

We now turn to the question of analyzing what equilibria look like in these games. Clearly, everyone choosing either A or B is a PNE (A is the "better" equilibrium, but B is still a PNE). But there are other equilibria as well, such as the ones illustrated in Figure 5.1.

Intuitively, the reason why these "non-trivial" equilibria arise is that there is some network structure that "blocks" the better action A from spreading

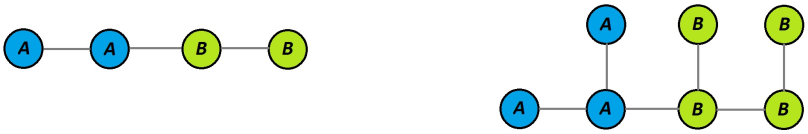


Figure 5.1: Illustrations of equilibria in a symmetric networked coordination game. The left example is an equilibrium when $x = y$; the right example is an equilibrium when $\frac{x}{2} < y < x$.

to more nodes in the network. We will return to an analysis of this phenomenon in Section 5.6; 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 converges if and only if a game has an ordinal potential function. Here we will consider the *social welfare* function given by

$$\Phi(\vec{a}) = \mathrm{SW}(\vec{a}) = \Sigma_{i \in n} u_i(\vec{a})$$

Notice that by expanding out the definition of u, we get

$$\Phi(\vec{a}) = \Sigma_{(i,j) \in E} Q(a_i, a_j) = 2\Sigma_{(i,j) \in E, i > j} Q(a_i, a_j) \tag{5.1}$$

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

Claim 5.2. *Φ is an ordinal potential function for symmetric networked coordination games.*

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

$$u_i(a'_i, a_{-i}) > u_i(a_i, a_{-i}) \tag{5.2}$$

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 5.1, the difference in the potential is $2\Sigma_{j \in N(i)}(Q(a'_i, a_j) - Q(a_i, a_j)) = 2(u_i(a'_i, a_{-i}) - u_i(a_i, a_{-i}))$ which by Equation 5.2 is strictly positive. ■

Thus, by Theorem 4.2 (which proved that BRD converges if and only if the game has an ordinal potential function), we directly get the following theorem.

Theorem 5.3. *BRD converges in every Symmetric Networked Coordination Game.*

Let us also remark that social welfare as an ordinal potential function is interesting in its own right. It means that whatever outcome we start off it, if people deviate to make themselves better off, they also make the outcome "better for the world". In addition, notice that if we start out with an outcome that maximizes social welfare, BRD will stay there; this, however, is not particularly surprising: under the assumption that $x > y$, there is a unique outcome that maximizes SW (namely, everyone choosing A) and this outcome is a PNE. Thus BRD will stay there; furthermore, if $x = y$, there are two outcomes (everyone choosing A or everyone choosing B), both of which are equilibria.

5.3 General Networked Coordination Games

So far, in this chapter, we have assumed that the coordination game is *symmetric*; in particular, this implies that everyone has the same "intrinsic value" for each product. In reality, some people have higher intrinsic value for different products (e.g. in the absence of network effects, some people naturally prefer iPhone, and others Android).

To model this, we can consider a more general class of games where

$$u_i(\vec{a}) = R_i(a_i) + \sum_{j \in N(i)} Q(a_i, a_j) \tag{5.3}$$

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 5.1, except that we now also parametrize the game by a intrinsic value function $R_i : \{A, B\} \to R$ for each node i and using equation 5.3 to define utility.

Notice that nodes can no longer employ the *same* simple decision rule of checking whether the fraction of its neighbours playing A exceeds some fixed threshold t—rather, each node has its own **subjective adoption threshold** $t(i)$ which depends on the number i's neighbors and its intrinsic value $R(i)$ 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 5.2 with $d + 1$ nodes, consisting of a "central" node v connected to d neighbors.

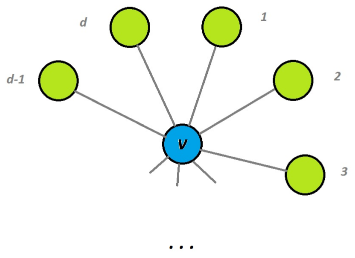


Figure 5.2: The star graph described in the example, with the equilibrium state highlighted (A in blue, B in green).

- Let $W_v(A) = d + \epsilon, W_v(B) = 0$.
- For all other nodes v', $W_{v'}(A) = 0$ and $W_{v'}(B) = 1 + \epsilon$.
- Let $Q(A, A) = Q(B, B) = 1$ (i.e. $x = y = 1$ in the game as described above).

So every node's coordination game indicates no network preference between A and B. But v "intrinsically" strongly prefers A, while all other nodes "weakly" prefer B.

What happens if everyone chooses A in this game?

- v has maximized intrinsic utility $(d + \epsilon)$ and maximized coordination utility (1 on each of d edges, d total), so there is no incentive to switch.
- Other nodes currently have no intrinsic utility and 1 coordination utility, so it benefits them to switch to B instead, receiving $1+\epsilon$ intrinsic utility and losing 1 coordination utility.
- This state has social welfare $3d + \epsilon$, but isn't an equilibrium.

What happens if everyone chooses B?

- Neighbors of v have maximized 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 coordination utility (1 on each edge), so it benefits v to switch to A instead, receiving $d + \epsilon$ intrinsic utility and losing d coordination utility.
- This state has social welfare $3d + d\epsilon$, but also isn't an equilibrium.

In fact, using best-response dynamics, we will *always* end in a state where v chooses A and other nodes choose B. This follows from the fact that A is strictly dominant for v and B is strictly dominant for all the other nodes; thus v playing A and everyone else playing B 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 $d+\epsilon$ 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 outcome which maximizes social welfare (which is attained in the "coordination" states where everyone plays either A or B). Furthermore, notice that this gap is 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 still converges?

Social welfare is no longer an ordinal potential function; as a counterexample, take the state in the graph above where all nodes choose B. If v switches to A, 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 $\Phi'(\vec{a}) = \sum_{(i,j)\in E, i>j} Q(a_i, a_j) + \sum_{i \in V} R_i(a_i)$. That is, we sum coordination utilities over every edge *once* (rather than twice as in social welfare; see Equation 5.1), and add the intrinsic values for each node.

Theorem 5.4. *BRD converges in every Networked Coordination Game.*

Proof. Let Φ' be above-described function. 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 games between i and $N(i)$ are affected by changing a_i to a_i', as well as the intrinsic value for i. So

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

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

■

5.4 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 social welfare in an equilibrium and *maximum social welfare*, $MSW = \max_{a \in A} SW(a)$, be? We now show that the gap between the SW of the *best* PNE and the MSW—referred to as the *"Price of Stability"*—cannot be too big.

Theorem 5.5. *In every Networked Coordination Game, there exists a PNE* $\vec{a}'$ *such that*

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

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 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 5.4 it will always converge, and this final outcome is a PNE. While SW may decrease at each step, as shown in the proof of Theorem 5.4 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. ■

5.5 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, 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 5.4 and Theorem 5.5) directly extend to this more general model, by defining $\Phi'(\vec{a}) = \Sigma_{(i,j)\in E, i>j} w_{i,j} Q(a_i, a_j) + \Sigma_{i \in V} R_i(a_i)$.

Note, however, that a players decision whether to play A is no longer just a function of the fraction of its neighbors playing A; rather, we now need to consider a **weighted subjective threshold** $t(\cdot)$ where node i switches to A whenever the fraction of its neighbors j *weighted* by $w(i,j)$ exceeds t.

5.6 Contagion

Let us now return to the question of how the adoption of a (better) 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 symmetric model (without intrinsic values and without weighted edges), but note that our analysis easily extends also to the more complex models.

Recall that in the symmetric model, a node decides to choose A if the fraction of its neighbors choosing A exceeds the "adoption threshold" $t = \frac{y}{x+y}$. We are interested in the question of when A will spread to the entire network; as we observed in Figure 5.1, there exist networks with equilibria where both A and B are played (and thus, even if players best respond, A 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 A so that they will choose A *no matter what*, will A 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 5.6. Given a symmetric networked coordination game 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 G if the following process *always* ends with *all* nodes in V choosing A:

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

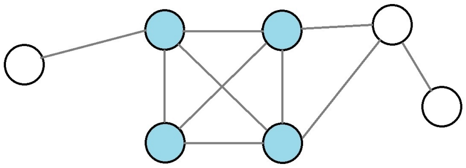


Figure 5.3: A graph with a $\frac{3}{4}$-dense set highlighted in blue. For every node in this set, at least $\frac{3}{4}$ of its neighbors also lie in the set.

Note that since BRD *always* converges in 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.

We now show that, quite surprisingly, there is a simple and elegant condition that exactly characterizes when a set S is cascading. To do this, we will define a notion of the *density* of a set S in a graph.

Definition 5.7. Given a 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 s'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 5.8. *Given a symmetric networked coordination game G induced by $G' = (V, E), Q$ with adoption threshold t, a set S is cascading w.r.t. 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 $T \subseteq V$ of density $1 - t$. Consider the first round in BRD when some node $v \in T$ becomes "infected" (i.e., switching to A). At this point, all of T's neighbors play B and thus, by the density requirement, the fraction of its neighbors playing B is at least $1 - t$, and consequently at most a fraction t play A, which contradicts that v would switch.

The "if" direction: Assume for contradiction that no $(1 - t)$-dense set exists in $V \setminus S$, yet S is not cascading (i.e., not everyone always gets infected by A). 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 when the cascade was not complete and let T be the set of nodes playing B in this outcome. Since nodes in S never switch actions, $T \subseteq V \setminus S$. Additionally, T must have density $1 - t$: otherwise, 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 T play A), but this contradicts the fact that the cascading process had reached a final state. ■

An interesting consequence of the above theorem is the that the *order* of the players in the cascade process (i.e., the restricted BRD) is irrelevant in determining whether or not a set S 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 BRD. We can consider an even stronger notion of cascading where the early adopters only need to start off playing A, but then may themselves participate in BRD (including potentially switching to the choice B if many of their neighbors are playing it).

Definition 5.9. Given a symmetric networked coordination game 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 G if BRD from the outcome (A_S, B_{-S}) *always* ends with *all* nodes in V choosing A.

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

Theorem 5.10. *Given a symmetric networked coordination game G induced by $G' = (V, E), Q$ with adoption threshold t, a set S is strongly cascading w.r.t. G if a) S has density t, and b) 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 5.8, nodes in S will never change from playing A (as at least a fraction t of their neighbors are in S and hence playing A at all times). Hence, running BRD from (A_S, B_{-S}) is equivalent to running BRD from (A_S, B_{-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 5.8 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.

The Computational Complexity of Finding the Small Cascading Sets

Ideally, we would like to have a computationally efficient way of finding a small cascading (or strongly cascading) set. It turns out that this problem is computationally intractable (technically, NP-complete). 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).

Dealing with Subjective (Weighted) Thresholds

So far we have only considered *symmetric* coordination games. Let us turn to analyzing also more general ones. Recall that for the case of symmetric coordination game, *each* player decides to switch to A if the fraction of its neighbors choosing A 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 contaigon still apply to networked coordination games with weigthed edges if we replace density for weigthed density in the theorem stataments.

Chapter 6

Traffic Network Games

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

6.1 Definition of a Traffic Network Game

A **traffic network game** 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 $T_e(x_e)$ on the "road" e if x_e 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 of each player i is a path p from s to t. (We could also define a different action set for each player with different s, t pairs, if for instance different people are traveling to/from different places).
- 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's utility is:

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

  Note that, since this utility is negative, longer travel times are worse.

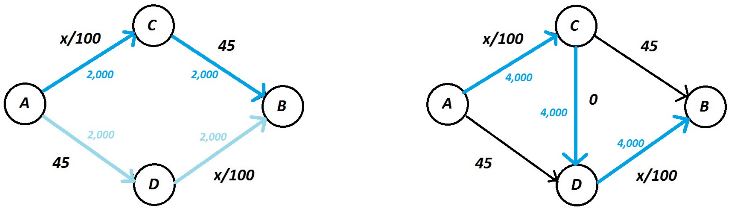


Figure 6.1: Left: A basic example of a traffic network game. Right: What happens to the PNE for this game when we add a road?

Example. Consider the traffic network in the left of figure 6.1 with 4,000 players moving from A to B, where there are four edges; $A \to C$ and $D \to B$ have travel time $\frac{x}{100}$, and $C \to B$ and $A \to D$ have travel time 45.

There are only two possible paths from A to B, and thus only two possible actions for the player: 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 85 $(40 + 45)$; 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 path with fewer players.

6.2 Braess's Paradox

Now, let us augment this network by adding an extremely efficient road from C to D with travel time 0. Intuitively, we would think that adding such an amazing new road would decrease the equilibrium state's 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 is the 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 following questions:

- Do equilibria always exist in traffic network games?
- Will BRD always converge in them?
- What is the price of stability in these games? (In other words, how much worse will selfish behavior make travel time?)

6.3 Convergence of BRD

As before, we will show that BRD converges, and hence that PNE exist, by developing a potential function for these games. First of all, let us look at the social welfare. Let $T(\vec{p})$ denote the *total travel time* of all players:

$$T(\vec{p}) = \sum_{i=1}^{n} \sum_{e \in p_i} T_e(x_e) = \sum_{e \in E} \sum_{i \in x_e} T_e(x_e) \tag{6.1}$$

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 (consider the game with the new highway, and the outcome where half the players play UP and half DOWN) this is not an admissible potential function (just as was the case with networked coordination games). Instead, similarly to networked coordination games, we can define a different potential function $\Phi(\vec{p}) = -L(\vec{p})$ where L is some variant "travel time energy" defined as follows:

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

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.

Claim 6.1. Φ *is an ordinal potential function for Traffic Network Games.*

Proof. Consider an action profile $\vec{p}$ and a player i who can improve their utility by playing p'_i. We wish to show that the "travel energy" L decreases (hence, Φ increases)—that is, that $L(p'_i, p_{-i}) - L(p_i, p_{-i})$ is negative. Observe that in this difference, only edges on p_i and p'_i are affected, and only on the disjoint edges (edges lying on one path but not both).

- On edges in $p_i \setminus p'_i$, we remove $\sum_{e \in p_i \setminus p'_i} T_e(x_e(\vec{p}))$.
- On edges in $p'_i \setminus p_i$, we add $\sum_{e \in p'_i \setminus p_i} T_e(x_e(\vec{p}) + 1)$.

But this is exactly the same as the change in i's travel time, which we know is negative because i's utility increases when changing paths. ■

So, it immediately follows (from the above and from Theorem 4.2) that:

Theorem 6.2. *BRD converges in all Traffic Network Games.*

6.4 Price of Stability

We now turn to analyzing the price of stability (i.e., the ratio between the best PNE's utility and the maximum social welfare).

Theorem 6.3. *In every Traffic Network Game, there exists a PNE* $\vec{p'}$ *such that*

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

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

Claim 6.4. *For every action profile* $\vec{p}$*,*

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

Proof. Clearly, by Equation 6.1, we have that $T(\vec{p}) \geq L(\vec{p})$. Additionally, we claim that $L(\vec{p}) \geq \frac{1}{2} T(\vec{p})$. Recall that

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

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

as desired. ■

So, just as in our coordination game proof (Theorem 5.5), pick some state $\vec{p}$ that maximizes $T(\vec{p})$, and run BRD until we arrive at a final state $\vec{p'}$; by Theorem 6.2 such a state must exist, and it is a PNE. Since L decreases at every step in this process (by Claim 6.1), we have

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

which concludes the proof. ■

We end this section by mentioning that an even stronger (and optimal) bound of $\frac{4}{3}$ was shown by Eva Tardos and Tim Roughgarden; their proof, however, requires more sophisticated machinery.

Chapter 7

Matching Markets

Let us now return to the second example in the introduction and to the bipartite matching problems we considered in chapter 3. Consider, for instance, the set of three people (A, B, and C) on the left side of a bipartite graph and the set of three houses (H_1, H_2, and H_3) on the right side, with edges between an individual and a house if that individual finds the house *acceptable*.

7.1 Defining a Matching Market

In general, given a set X of people, a set Y of houses, and a bipartite graph $G = (V = X \cup Y, E)$, recall that a matching M is a set of edges $M \subset E$ such that each node in G has degree at most 1 with respect to M.

- The *maximum matching* problem consisted of finding the largest such matching (in this context, the largest number of people who can get a house).

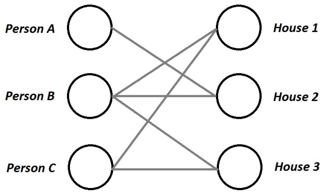


Figure 7.1: A basic example of a matching problem between people and houses.

- The *perfect matching* problem instead requires us to find a matching where every node is successfully matched (i.e. every person gets a house and every house is occupied).

But, in such matchings, we are only guaranteed that the people get some house they find "acceptable". In reality, people have *preferences* over houses; we represent these preferences by specifying a *subjective valuation* of each house. Additionally, let's assign prices to houses.

Specifically, given a set of players (buyers) $X = [n] = \{1, \dots, n\}$ 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 play); buyers that do not get any item get utility 0.

Without loss of generality, we can assume $|X| = |Y|$; we can always convert another setting to this case by adding "dummy buyers" (who value all items at 0) or "dummy items" (with value 0).

Definition 7.1. We refer to the tuple $\Gamma = ([n], Y, v)$ (as defined above) as a **matching market frame**, and the tuple $(\Gamma, p) = ([n], Y, v, p)$ as a **matching market**.

7.2 Acceptability and Preference

We can now 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).

Definition 7.2. Given a 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.

But the fact that y is merely 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 maximizes $v_i(y) - p(y)$ over all items in Y; that is

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

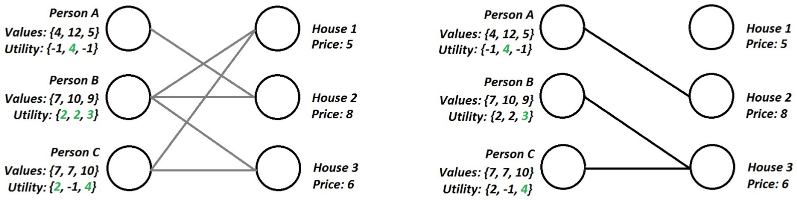


Figure 7.2: Left: An acceptability graph for an example matching market with the given prices and values. Buyers are linked to any houses for which they have non-negative utility. Right: The preferred choice graph for the same example. Note that now each buyer is linked only to houses for which they have the *highest* utility.

Note that there may not necessarily exist a unique preferred choice for each buyer; 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 7.3. Given a 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.

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!) But there may not exist a perfect matching in the preferred choice graph—for instance, it is relatively easy to construct a scenario where two buyers both prefer the same item regardless of its price.

A basic question to consider is the one presented in the introduction: is there a way to set prices so that everyone ends up with an outcome they are the *most happy* with? That is, can we set prices so that there exists a perfect matching in the preferred choice graph, and so everyone ends up with their preferred choice? The notion of *market clearing* (and market equilibrium) addresses this:

Definition 7.4. Given a matching market frame $\Gamma = ([n], Y, v)$, 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** (or assignment), and the pair (p, M) as a **market equilibrium** (or **Walrasian equilibrium**).

7.3 Social Optimality of Market Clearing

Whereas a matching in the acceptability graph by definition ensures that each buyer i who matched to an item y gets positive utility, it turns out that a matching M in the preferred choice graph ensures that the assignment of buyers to items *maximizes* the *social value* of the items.

Let us formalize this: An *assignment* $\alpha : X \to Y \cup \perp$ is a function that, given a buyer $i \in X$, produces either an item $y \in Y$ (meaning i was assigned item y) or $\perp$ (meaning that i was not assigned an item), such that for no two buyers $i, j \in X$ is it true that for some $y \in Y$, $\alpha(i) = \alpha(j) = y$. Note that any matching M induces such an assignment α_M, where $\alpha(i)$ is simply the item to which i is matched in M (or $\perp$ if i is not matched).

Now we can define the social value of an assignment in a matching market frame Γ as

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

Note that this is distinct from the *buyers' total welfare* (the the sum of their utilities), as it consists solely of the total *valuations* of the items, but without considering their prices. But if we include the sellers' utilities (i.e., the sum of all prices paid for items in M), then the total welfare of both buyers and sellers (i.e., the social value) is in fact equal to social welfare:

$$SW^\Gamma(\alpha, p) = \text{buyers' utilities} + \text{sellers' utilities}$$

$$= \sum_{i \in [n], \alpha(i) \neq \perp} u_i(\alpha(i)) + \sum_{i \in [n], \alpha(i) \neq \perp} p(\alpha(i))$$

$$= \sum_{i \in [n], \alpha(i) \neq \perp} (v_i(\alpha(i)) - p(\alpha(i))) + \sum_{i \in [n], \alpha(i) \neq \perp} p(\alpha(i))$$

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

We thus have the following claim:

Claim 7.5. *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 $SW^\Gamma(\alpha, p) = \max_{\alpha', p'} SW^\Gamma(\alpha', p')$.

Immediately by Claim 7.5 we have:

Corollary 7.6. *For any matching market frame $\Gamma = ([n], Y, v)$, an assignment α maximizes social value if and only if α, p maximizes social welfare for any (or every) p.*

So, if we have an outcome that maximizes social value, then social welfare will be maximized *regardless of how we set prices*!. However, in order to obtain such a favorable outcome, it turns out that prices (specifically, market clearing prices) are important:

Theorem 7.7 (Social optimality of market equilibria.)**.** *Given a matching market frame Γ and a market equilibrium (p, M) for Γ, we have that α_M maximizes social value.*

Proof. Consider some market equilibrium (p, M) for Γ and some assignment α' that maximizes social value for Γ. By Corollary 7.6, (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,

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

since each buyer does at least as well in α as in α', and all items are sold in α thus the sellers' utilities are maximized as well. And so (α, p) must also maximize social welfare, and, again by Corollary 7.6, α maximizes social value. ■

Let us end this section by noting that whereas the proof of Theorem 7.7 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 7.7 is an instance of what is known as "the first fundamental theorem of welfare economics" and a formalization of Adam Smiths famous "invisible hand" hypothesis.

7.4 Existence of Market Clearing Prices

So, given the awesomeness of market clearing prices, do they always exist? In fact, Shapley and Shubik (1972) proved that that they do! Thus, 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 the items' social value.

We will present a constructive proof, originally published by Demange, Gale, and Sotomayor in 1986, and furthermore show how the market equilibrium can be efficiently found (relying on the material covered in Chapter 3). This theorem is an instance of what is referred to as "the second fundamental theorem of welfare economics".

Theorem 7.8. *A market equilibrium (p, M) exists in every matching market frame 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 $p(y) = 0$ or all $y \in Y$. Next, update prices according to the following procedure:

- If there exists a perfect matching M in the preferred choice graph, we are done and can output (p, M). (Recall that, by Theorem 3.6, this can be done efficiently.)
- Otherwise, by Theorem 3.9, there exists a constricted set of buyers $S \subset [n]$ (i.e. a set such that $|N(S)| < |S|$) in the preferred choice graph (which may also be found efficiently).
- In this case, 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 illustration of this procedure can be found in Figures 7.3 and 7.4 Clearly, if this procedure terminates, we have found a market equilibrium. Using a potential function argument, we can argue that the 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*, i.e. $\max_y(v_i(y) - p(y))$. (Note that, if we have a perfect matching in the preferred choice graph, this is actually the utility buyer i gets, but in the general case there could be unmatched buyers.)
- Let $\Phi^B(p) = \sum_{i \in [n]} \Phi_i^B(p)$, the sum of the potential for all buyers.
- Let $\Phi^S(p) = \sum_y p(y)$, 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.)
- For the total potential, let $\Phi(p) = \Phi^B(p) + \Phi^S(p)$.

Claim 7.9. *After every price update, $\Phi(p)$ decreases.*

Proof. We first increase the price of the items in the neighborhood $N(S)$ of our constricted set 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 (note that we here rely on the fact that values are integers, or else there could be some other item that previously was not preferred by i, but now becomes preferred, and only costs e.g., 0.5 more) . 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.

Now, if we do shift all prices downwards by one at the end of the current iteration, Φ^S will decrease by n, but Φ^B will increase by n (by the same logic as above), and so there will be no further change to Φ. ■

Claim 7.10. *After every price update, $\Phi(p) \geq 0$.*

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 then follows that $\Phi^S(p) \geq 0$.

Next, since there always exists some item with price zero, 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 $\Phi(p) \geq 0$. ■

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 of 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. ■

Emergence of Market Equilibria

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 of how they might arise over time. If demand for a particular item is high (there are too many people who prefer that item, i.e. a constricted set in the preferred choice graph), then the market will adjust to increase the price 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

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!

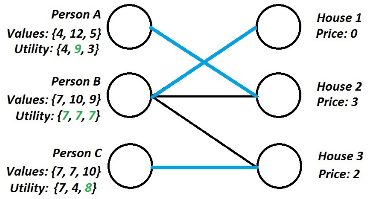


Figure 7.3: The algorithm detailed here when applied to the matching market above. The perfect matching on the final preferred-choice graph (i.e. the equilibrium assignment) is shown in blue.

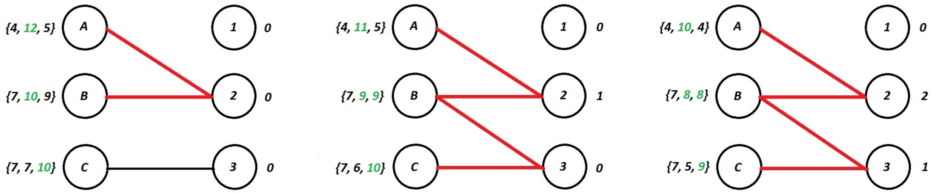


Figure 7.4: The preferred choice graphs for the intermediate states of the algorithm. Constricted sets are indicated by red edges.

too expensive for some buyer (in other words, when the potential utility of some buyer becomes negative) and thus we can't sell everything off. In such a case, it may seem natural for the sellers to decrease market prices.

In general, these type of market adjustment mechanisms, and variants thereof, are referred to as *tatonnement* (French for "groping") 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

To conclude, let us remark that market equilibria are not necessarily unique. Some equilibria are better for the seller, whereas others are better for the buyers. (In particular, we can often shift prices by a small—or, as in Figure

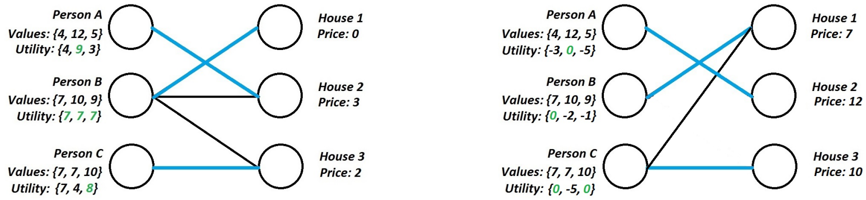


Figure 7.5: Left: The equilibrium we found with the algorithm is, in fact, buyer-optimal. Right: If we increase the prices of all houses by as much as possible such that our matching in the preferred choice graph stays the same (even though the graph itself doesn't!), we obtain the seller-optimal equilibrium. Notice that the seller now gets all of the utility!

7.5, 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.)

7.5 Bundles of Identical Goods

Finally, let us consider a simplified special case of the matching market model, wherein 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 by decreasing size, i.e. $c_1 \geq c_2 \geq \ldots \geq c_n$.

Observe now 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. By similar logic, the person who values the good the second-most should get y_2, and so on. Hence, by Theorem 7.7, this property must also hold in any market equilibrium.

We will now show that in any market equilibrium, the largest bundles will also actually have the highest price per object.

Theorem 7.11. *For every matching market frame for bundles of identical goods, given by* $\Gamma = (n, Y, v^{\vec{c},\vec{t}})$*, and for every market equilibrium* (p, M) *for such a* Γ*, it is the case that, for every* $j = 1, \ldots, n-1$, $\frac{p(y_j)}{c_j} \geq \frac{p(y_{j+1})}{c_{j+1}}$.

Proof. Consider a market equilibrium (p, M). Let α_j be the price per object $(\frac{p(y_j)}{c_j})$ in bundle j. Consider two items j, j' where $j < j'$, and assume for the sake of contradiction that $\alpha_j > \alpha_{j'}$, i.e. the items in the *larger* bundle y_j are also *cheaper.*

Now consider the player i who is matched to the smaller bundle $y_{j'}$ of more expensive goods. His utility is currently $c_{j'}(t_i - \alpha_{j'})$. But 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 (implying $t_i - \alpha_j > t_i - \alpha_{j'}$). So j' cannot be a preferred choice for i, contradicting the fact that this is a market equilibrium and proving the theorem. ■

At first glance, this result may seem counter-intuitive, as the "unit price" for buying a larger number of items in this situation is actually larger; usually, in situations such as grocery shopping, the unit price tends to decrease when we buy a larger quantity of something!

However, the reason why this occurs is because the supply of bundles is limited here, and each buyer can only purchase a single bundle. In such a setting, it makes sense that the price per object would increase as you buy more; specifically, as long as the "unit price" is lower than your value, your utility will increase as the bundle size does, and so intuitively people should be willing to pay a higher unit price for larger bundles.

Chapter 8

Exchange Networks

Let us consider now a more generalized version of matching markets, known as **exchange networks**.

8.1 Defining Exchange Networks

- Assume we have a graph $G = (V, E)$ with n nodes (the players).
  - In the matching market scenario, we required G to be bipartite and have a set of n buyers and n items.
- Each node may participate in *at most one* "exchange" with any one of its neighbors; this models the idea of nodes forming "exclusive partnerships".
  - In the matching market scenario, the "exchange" is analogous to the matching between a buyer and an item, and is similarly one-to-one.
- Each edge e has an amount of "money" $v(e)$ that can be split between its endpoints if and only if an exchange between the endpoints takes place. (This need not be an even split, either.)
  - In the matching market scenario, edges are between buyers and items, and we can think of the "price" of an edge as the valuation of that buyer for that item.
  - The buyer's cut of the profit from the exchange is equal to the utility he derives from the sale (his value minus the price).

  - The seller's cut corresponds to the remaining portion (the price of the item).

We can define an exchange network formally, as follows.

Definition 8.1. An **exchange network** is defined by a pair (G, v), where $G = (V, E)$ is a graph and $v : E \to \mathbb{N}$ is a function.

Definition 8.2. 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. (This is analogous to our earlier definitions of a matching, except that G need not be bipartite).
- $a : V \to \mathbb{R}^+ \cup \{0\}$ is a function representing the allocation of edge values to nodes such that, for every edge $e = (u, v) \in M$, $a(u) + a(v) = v(e)$, and for every node v not part of an edge in M, $a(v) = 0$.

Mathematically, this is a generalization of matching markets from bipartite graphs to general graphs. But conceptually, even if our particular choice of G is bipartite, we no longer need to think of the nodes in this graph as "buyers" and "items"; instead, we now have a social network of nodes, and of particular interest is which nodes in this network are "more powerful" than other nodes.

8.2 Stable Outcomes

We call an outcome *stable* if no node can improve its own allocation of the edge values by proposing a new "offer" to one of its neighbors that improves both nodes' allocations (and *strictly* improves at least one node's allocation). Formally:

Definition 8.3. An outcome (M, a) is **unstable** if there exists a pair of nodes x, y such that $(x, y) \notin M$, but $a(x) + a(y) < v(x, y)$. Any outcome that is not unstable is **stable**.

In particular, if $a(x)+a(y) < v(x, y)$ (i.e. the two nodes' current allocations are smaller than what they would get combined by forming a partnership), we can define this *surplus* to be $s = v(x, y) - a(x) - a(y)$. Then we can propose a new partnership with allocation $a'(x) = a(x) + s/2$ and $a'(y) = a(y) + s/2$, which will be favorable to both x and y. Conversely, if x and y have some allocation such that they would prefer to form a partnership to their current arrangements, then $v(x, y)$, the value of their partnership, must be greater than the sum of their current allocations $a(x) + a(y)$.

Now we will show that, if G is bipartite, the stability of an exchange network outcome is a generalization of market clearing.

- Given a matching market frame $\Gamma = ([n], Y, v)$, where the range of v is $\mathbb{N}$, we say that (G, v') is the exchange network corresponding to Γ if:
  - $G = (V, E)$ 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).
  - $v'(i, y) = v_i(y)$ (buyer-to-item valuations are the same).
- Given a bipartite exchange network $(G = (([n] \cup Y), E), v')$, we say that $\Gamma = ([n], Y, v)$ is the matching market frame corresponding to (G, v') if:
  - $v_i(y) = 0$ if $(i, y) \notin E$
  - $v_i(y) = v'(i, y)$ otherwise (if $(i, y) \in E$).

In general, in a bipartite exchange network $(G = (([n] \cup Y), E), v')$, there are no designated "buyers" and "seller", and nodes in X and Y are treated symmetrically. In contrast, the matching market setting always assigns X to be the buyer; hence, by declaring either X or Y to be the buyers $[n]$, we can construct *two* matching frames corresponding to each bipartite exchange network. Without loss of generality, we will assume that X is the set of buyers in all cases and focus on bipartite graphs $G = ([n] \cup Y, E)$.

Claim 8.4. *Let $\Gamma = ([n], Y, v)$ be a matching market frame and let (G, v) be the exchange network corresponding to Γ. Or, conversely, let $(G = ([n \cup Y], E), v)$ be a bipartite exchange network, and let Γ be the matching market frame corresponding to it. Then (M, p) 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. Assume (M, p) is a market-clearing equilibrium, and assume for contradiction that (M, a) is not stable, i.e. there exists some unmatched buyer-item pair (i, y) such that $a(i) + a(y) < v(i, y)$; that is, by our construction of a,

$v_i(M(i))p(M(i))+p(y) < v_i(y)$, and so $v_i(M(i))p(M(i)) < v_i(y)-p(y)$. Then i prefers item y to his matching $M(i)$, contradicting (M, p) being a market-clearing equilibrium.

Conversely, assume that (M, a) is stable, but (M, p) is not market-clearing. Then there exists some buyer i that prefers getting 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 $v_i(M(i))p(M(i)) < v_i(y) - p(y)$, implying $v_i(M(i))p(M(i)) + p(y) < v_i(y)$; by our construction of a, $a(i) + a(y) < v(i, y)$, and we arrive at the contradiction that (M, a) cannot be stable.

The only other case is $M(i) = \perp$. Then $a(i) = 0$, but, since i prefers y, $v_i(y) > p(y) = a(y)$. So $a(i) + a(y) = a(y) < v_i(y) = v'(i, y)$, once again contradicting (M, a) being stable. ■

Combining this result with the theorem (7.8) that a market-clearing equilibrium always exists in a matching market, we directly obtain:

Corollary 8.5. *If G is bipartite, then the exchange network (G, v) has a stable outcome.*

However, this is not the case when G is not bipartite. In fact:

Claim 8.6. *There exists an exchange network with three nodes for which no stable outcome can exist.*

Proof. (By construction.) Assume a "triangle" with three nodes and every pair of nodes connected by an edge. Let all edges have the same weight k. Any matching M must contain exactly one edge. Assume there is such a matching $M = \{(u, v)\}$ which defines a stable outcome; at least one of the two nodes $x \in \{u, v\}$ must have $a(x) < k/2$. Then, for the currently unmatched node w in the graph, it is clear that $a(x) + a(w) = a(x) < k/2 < k = v(x, w)$. This matching is clearly unstable. ■

In fact, in the example above, we can always propose an exchange that has surplus $k/2$ (and so, if we split the surplus evenly enough, e.g. if w offers x $3k/4$ and himself $k/4$, both players will always gain at least $k/4$ from deviating). This in particular rules out even weakened notions of stability where people will not deviate unless it makes them *substantially* better off (i.e. by some constant ϵ).

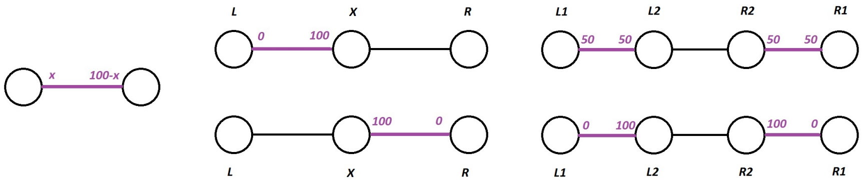


Figure 8.1: Some basic examples of exchange network games, with stable matchings/bargains indicated in purple.

8.3 Bargaining in Real Life

Let us now observe some examples; consider the following graphs in figure 8.1. Let $v(e) = 100$ for all of these (real-life experiments often focus on the model of having people split a dollar, or 100 cents).

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 return to this point soon.)

Next, the second example, with three nodes in a line, has only two stable outcomes. The middle player X can choose to match with L or R, but in any stable outcome *he must take all 100 units of the split.* Otherwise, the player (L or R) to which X is not matched can propose an acceptable arrangement in which X takes slightly more than in his current arrangement.

In the final example, any stable outcome is characterized by two matchings (L_1, L_2) and (R_2, R_1). (Notably, if a stable outcome contains (L_2, R_2), one of the two will have a cut of less than 50 from this arrangement, and the corresponding player L_1 or R_1 can offer them a better deal. So no stable outcome contains the middle edge, meaning that every stable outcome must contain the other two.) However, any outcome using these edges where L_2's cut and R_2's cut sum to more than 100 will be stable—for instance, the two outcomes highlighted in the figure, one in which every player gets 50 and one in which L_2 and R_2 both take the whole 100.

An important realization about these examples is that nodes on the "ends" of these 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 a small amount of power. Consider the following **ultimatum game** to model this situation:

- There are two players, A and B.
- A begins by proposing any split of one dollar between himself and B.
- B can either *accept* the split and take the utility 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 equilibrium of this game is for A to propose that he takes the whole pot and for B to accept!*

This is clearly unrealistic, since nobody would accept such an unfair deal in real life. Usually, there is some "cutoff" 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 cutoff has been found to be cultural and to depend on several different factors.)

We can model this phenomenon by adding a "moon" node x_m connected only to x, where the edge (x, x_m) has weight equal to x's cutoff value. So x would never accept any offer where he earns less than $v(x, m_x)$, since it would obviously just rather match with m_x.

8.4 Fairness and Balanced Outcomes

As seen in even our example two-node graph, where anything was a stable outcome, stability does not provide very much predictive power. We would like to formalize a stronger notion to predict what is most likely to happen—for instance, in the two-node graph, we would expect to see something close to an even split.

Consider a two-node network with players A and B and edge weight one. But now let us 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 option with utility y. Let the surplus that the two players earn from bargaining with one another instead of taking their outside options be denoted by $s = 1 - x - y$.

Clearly, A will never accept any ofer that gives him less than x, and B will never accept any offer where he earns less than y. So they are effectively now

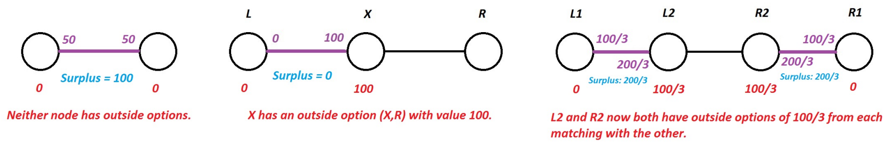

50
50
Surplus = 100
0
0
Neither node has outside options.
L
X
R
0
100
Surplus = 0
0
100
X has an outside option (X,R) with value 100.
L1
L2
R2
R1
100/3
200/3
Surplus: 200/3
100/3
200/3
Surplus: 200/3
0
100/3
100/3
0
L2 and R2 now both have outside options of 100/3 from each matching with the other.

Figure 8.2: Returning to the examples in figure 8.1, 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.

negotiating over how their surplus s should be split; effectively, this brings us back to the simple two-node example, where we would expect, by our notion of "fairness", the surplus to be split evenly.

So A should receive $x + s/2$, and B should receive $y + s/2$; this amounts to a proper split of $v(x, y)$ that is similarly attractive to both players over their outside options. John Nash, in his results about bargaining, suggests that this is in fact the "most fair" way to set prices.

We can in fact generalize this to general exchange networks; we can say that an outcome in such a network 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. More precisely, define the outside option of a node x to be $\max_{y \neq M(x)}\{v(x, y) - a(y)\}$, the maximum value it can earn by making an acceptable offer to some player besides the one it is currently matched with.

Balanced outcomes provide good predictions of how human players will actually split money, not only in the case of a two-node network, but also in more general networks. In fact, there is an elegant result by Kleinberg and Tardos which states that every exchange network that has a stable outcome also has such an outcome that is balanced.

Chapter 9

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 will now turn to a single-round version of this problem, where we have a set of buyers, a set of items for sale, and this time a *single seller* (e.g. Google selling slots for sponsored search advertisements).

We are interested in creating *mechanisms* where buyers first report their values for different items, and then, based on what the buyers report, the seller produces an assignment and a set of prices. In particular, we are interested in mechanisms that *ensure an efficient allocation* of items to buyers—that is, we wish to ensure an allocation of items that maximizes social value and thus, by Corollary 7.6, also social welfare. (As an aside, the seller may instead choose to maximize 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" seller and thus only consider social value maximization.) To consider this question, and generalizations of it, let us begin by introducing a framework for mechanism design.

9.1 The Mechanism Design Model

Consider a scenario where we have:

- A set of n players.
- A finite set of states X.
- Each player i is associated with a *type* t_i in some finite *type space* T.

- Each player i is associated with a *valuation function* $v_i : T \times X \to \mathbb{N}$, where $v_i(t_i, x)$ describes how a player i having type $t_i \in T$ values a state $x \in X$.

We refer to such a tuple (n, X, T, v) as a **social choice context**, or simply a **context**.

Let us now consider the following process induced by a mechanism M:

- Each player i submits a "report" r_i to the mechanism 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 $x \in X$ and a profile of prices/payments $\vec{p} \in \mathbb{R}^n$; we refer to the tuple $(x, \vec{p})$ as the **outcome**.
- The utility of a player i in an outcome $(x, \vec{p})$ is then given by

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

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. This utility function clearly 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 social context and the above-described process, let us consider some basic examples, starting with the fundamental concept of an auction.

Example: First-price auction. Consider a scenario where:

- We have a set of n buyers, and a single item for sale.
- The states $X = [n] \cup \perp$ determine which of the n players will receive the object (where $\perp$ represents no player winning).
- Player i's type t_i is i's valuation of the object, and $v_i(t_i, x) = t_i$ if $x = i$ (i.e., i gets the object) and 0 otherwise.

- Players report their valuation of the object (i.e. their type), r_i, to the mechanism.
- The mechanism returns $M(\vec{r}) = (i^*, \vec{p})$, where:
  - $i^* = \text{argmax}_i\{r_i\}$ (i.e. the winner is the player who reports (or "bids") the highest value).
  - $p_i = 0$ 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 however much the *second-highest* bidder bid; that is,

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

Goals of Mechanism Design

The goal of mechanism design is to, given some social choice context $\Gamma = (n, X, T, v)$, design a mechanism 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 context 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

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

Definition 9.1. Given a context $\Gamma = (n, X, T, v)$ and a type profile $\vec{t} \in T^n$, we say that a state x **maximizes social value** if $x = \text{argmax}_{x \in X} SV^{\Gamma}(\vec{t}, x)$.

We can also define social welfare as we did in Chapter 7 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 context of an auction); by the same argument as in Claim 7.5, 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 using concepts of game theory, we must first view the process as a game: Given a context $\Gamma = (n, X, T, v)$ a type profile $\vec{t} \in T^n$, and a mechanism M, let $G^{\Gamma,\vec{t},M}$ denote the game induced by $\Gamma, \vec{t}$, and M, wherein each player i chooses some report r_i and their utility is defined as above based on the state and prices ultimately chosen by M.

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

Definition 9.2. A mechanism M is **dominant-strategy truthful (DST)** for the context $\Gamma = (n, X, T, v)$ if, for every $\vec{t} \in T^n$, t_i is a dominant strategy for player i in $G^{\Gamma,\vec{t},M}$.

So, if M is DST, then rational players will always report their true type (e.g., for auctions, their true valuation of the object) rather than potentially lying about it, *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 all players reporting their types truthfully is a Nash equilibrium.

Definition 9.3. A mechanism M is **Nash truthful** (NT) for the context $\Gamma = (n, X, T, v)$ if, for every $\vec{t} \in T^n$, $\vec{t}$ is a Nash equilibrium in $G^{\Gamma,\vec{t},M}$.

As it turns out, these notions are equivalent:

Claim 9.4. *A mechanism 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 $\vec{r_{-i}}$ 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 $\vec{t'} = (t_i, r_{-i})$; this contradiction proves the claim. ■

(As an advanced aside, the literature on mechanism design often also considers a setting, referred to as *Bayesian*, where players' types come from a probability distribution D. Then we may consider an even weaker notion of a "Bayes-Nash truthful" mechanism, where every player's *expected* utility is maximized by correctly reporting their type. This notion is no longer equivalent to DST.)

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

Definition 9.5. A mechanism M is said to **DST-implement social value maximization** for the context $\Gamma = (n, X, T, v)$ if M is DST for Γ and, for every $\vec{t} \in T^n$, $M(\vec{t})$ 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*, a notion similar to DST-implementation which for brevity we will not discuss here), which only requires the existence of *some* Nash equilibrium (not necessarily a truthful one) that leads to a social-value-maximizing outcome:

Definition 9.6. A mechanism M is said to **Nash-implement** social value maximization for the context $\Gamma = (n, X, T, v)$ if, for every $\vec{t} \in T^n$, there exists a PNE $\vec{t'}$ for $G^{\Gamma,\vec{t},M}$ such that $M(\vec{t'})$ maximizes social value with respect to Γ and $\vec{t}$.

Revisiting 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 far 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 our second-price auction, we have the following nice theorem.

Theorem 9.7. *The second-price auction DST-implements social value maximization.*

Proof. We argue that no player can ever improve their utility by either overbidding or underbidding, as follows:

Overbidding: By bidding your true valuation, you clearly can never end up with negative utility (either you lose and get nothing, or you win with the second price equal to or less than your value; either way your utility is non-negative).

- If your value is the highest bid, and you increase your bid, then the amount you pay (and hence your utility) will not change; you will still get the item and pay the second price.
- If your value is less than the highest bid, then, unless you set your bid equal to or above the highest bid, your utility will not change (as you will get nothing). If you do set it equal to or above the highest bid, then that bid will become the new second price, and you will end up buying the item at a price equal to the old highest bid, thereby ending up with negative utility (as your true value is lower).

Underbidding: Again, by bidding truthfully, you must end up with non-negative utility.

- If your value is less than the highest bid, and you decrease your bid, then you will still get nothing; your utility will not change.
- If your value is greater than or equal to the second price (i.e. you have the highest bid), then keeping it greater than or equal to the highest bid will not affect your utility, as you will still buy the item for the second price. If you bid below the second price, you will get nothing and end up with zero utility, which cannot be better than bidding truthfully.

So second-price auctions are DST. Finally, by construction, if everyone bids truthfully, then the player who values the item the most will win it, thus maximizing social value. So second-price auctions DST-implement social value maximization. ■

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

Theorem 9.8. *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 $r_{i^*} = t_{i^{**}}$. Players besides i^* can only receive negative utility by deviating from their true valuation. Meanwhile, i^* (the winner of the item) loses utility from overbidding, and receives zero (loses the item) from underbidding. So this is a PNE. Additionally, since the object is still assigned to the player with the most value for it, this implements social value maximization. (Perhaps surprisingly, notice that this Nash equilibrium pricing scheme is identical to a second-price auction!) ■

Notice, however, that finding the Nash equilibrium in the case of a first-price auction requires the players to know each others' valuations. While this may be reasonable to assume in a world where the auction is run repeatedly, 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 bid (underbid) based on their beliefs about the other players' valuations. Formalizing this requires assumptions about the distributions of players' valuations, and requires using the more general concept of a Bayes-Nash equilibrium; we will omit the details in this course.

9.2 The VCG Mechanism

A natural question following the above is that of whether we can find a mechanism that implements social-value maximization in *any* (general) social choice context. In fact, the amazing Vickrey-Clark-Groves (VCG) mechanism shows that this is in fact possible; in particular, as we shall later see, we can use this mechanism to design an auction for matching markets.

Theorem 9.9. *For every social choice context Γ, there exists a mechanism M that DST-implements social value maximization for Γ.*

Proof. (By construction.) Given a context $\Gamma = (n, T, v)$, the VCG mechanism $M(\vec{r})$ outputs an outcome $(x^*, \vec{p})$ such that:

- $x^* = \text{argmax}_{x \in X} \text{SV}^\Gamma(x, \vec{r})$.
- $p_i = -\sum_{j \neq i} v(r_j, x^*)$. (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 x^* 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 outcome $(x^*, \vec{p})$, player i will obtain utility equal to $v(t_i, x^*) - p_i = v(t_i, x^*) + \sum_{j \neq i} v(r_j, x^*) = SV^\Gamma(x^*, (t_i, r_{-i}))$. So player i should submit a report r_i such that M chooses x^* to maximize this expression. But, by submitting t_i, M will by construction pick such a state. So submitting $r_i = t_i$ is dominant; hence M is DST and DST-implements social value maximization. ■

Computational efficiency of the VCG mechanism. Note that if we can find the socially optimal outcome, then the VCG mechanism is efficient. For

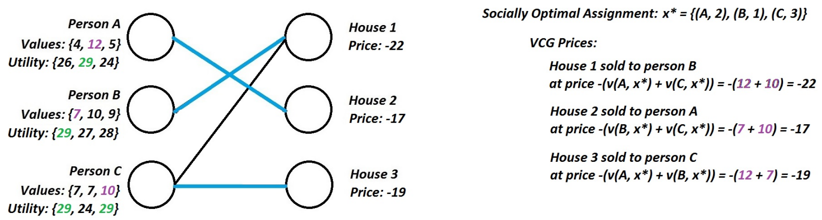


Figure 9.1: An illustration of standard VCG pricing without the Clark pivot rule. The price each player "pays" is the negative of the other players' social value in the socially optimal equilibrium.

many problems, however, it is hard to find the socially optimal state (even if people truthfully report the preferences); consequently, the CS literature has focused on studying *approximation algorithms*, where we find a close-to-optimal state. In some cases, we can design mechanisms that achieve similar guarantees, however it can be shown (assuming the existence of secure encryption schemes) that a mechanism such as VCG that simply uses the underlying algorithm as a black-box cannot exist [Pass-Seth 2013, Chawla-Immorlica-Lucier 2013].

The Clark pivot rule: paying your "externality". In the VCG mechanism described above, the operator has to pay the players. However, it is fairly straightforward to modify the mechanism so that the players pay the operator. Specifically, if we shift the price of player i in a manner independent of i's report:

$$p_i = V(r_{-i}) - \sum_{j \neq i} v(r_j, x^*)$$

for some function $V(r_{-i})$), the mechanism will remain DST, since such a price adjustment cannot affect i's preference between actions.

A particularly useful DST-preserving adjustment is the following; let,

$$V(r_{-i}) = \max_{x \in X} \sum_{j \neq i} v(r_j, x)$$

This effectively computes the state that would maximize social value if we were to exclude player i from the game. This is referred to as the *Clark pivot*

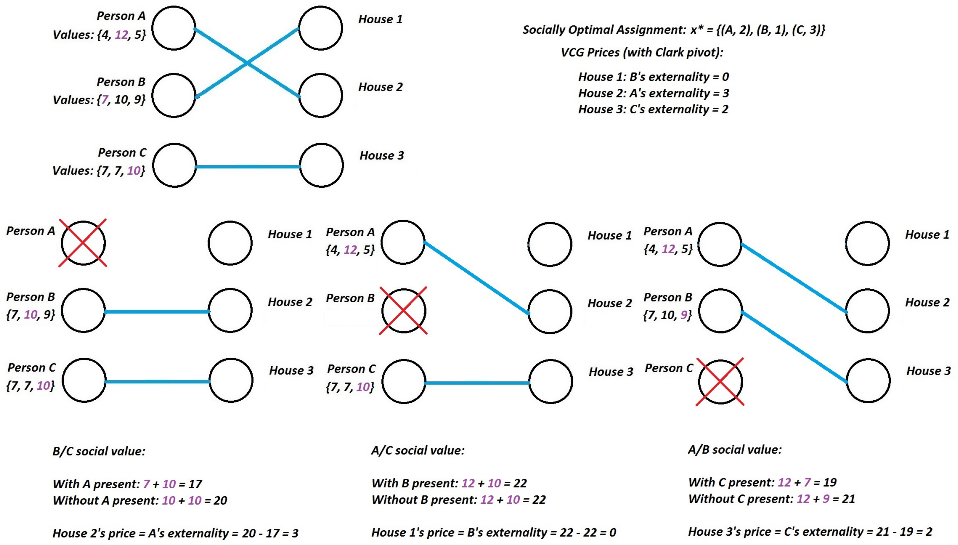

Person A
Values: {4, 12, 5}
Person B
Values: {7, 10, 9}
Person C
Values: {7, 7, 10}
House 1
House 2
House 3
Socially Optimal Assignment: x* = {(A, 2), (B, 1), (C, 3)}
VCG Prices (with Clark pivot):
House 1: B's externality = 0
House 2: A's externality = 3
House 3: C's externality = 2
Person A
Person B
{7, 10, 9}
Person C
{7, 7, 10}
House 1
House 2
House 3
Person A
{4, 12, 5}
Person B
Person C
{7, 7, 10}
House 1
House 2
House 3
Person A
{4, 12, 5}
Person B
{7, 10, 9}
Person C
House 1
House 2
House 3
B/C social value:
With A present: 7 + 10 = 17
Without A present: 10 + 10 = 20
House 2's price = A's externality = 20 - 17 = 3
A/C social value:
With B present: 12 + 10 = 22
Without B present: 12 + 10 = 22
House 1's price = B's externality = 22 - 22 = 0
A/B social value:
With C present: 12 + 7 = 19
Without C present: 12 + 9 = 21
House 3's price = C's externality = 21 - 19 = 2

Figure 9.2: An illustration of VCG pricing using the Clark 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 Clark VCG price for a player is equal to the standard VCG price plus the other players' social value with that player absent, independent of what that player bids.

rule. Note that using this rule ensures that the price:

$$p_i = \max_{x \in X} \sum_{j \neq i} v(r_j, x) - \sum_{j \neq i} v(r_j, x^*)$$

is always non-negative, since the social value excluding i at any state x^* can clearly never be higher than the maximum social value excluding i. So players will always pay the operator using this price adjustment.

The interpretation of the VCG mechanism with this particular rule added is that each player i pays for the "harm" in value that 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 $\sum_{j \neq i} v(r_j, x^*)$ value; without i present, they would get $\max_{x \in X} \sum_{j \neq i} v(r_j, x)$. In other words, p_i is given by (social value for other players if i is absent) - (social value for other players with i present), where these social values assume that types are reported truthfully. Let's look at a brief example of a VCG mechanism in terms of these externalities.

Example: "Voting with money transfers." Let's say that we have two voters and our set of outcomes X is {Clinton, Trump}. Let the valuations of each voter be given by $v_1(\text{Clinton}) = 10$, $v_1(\text{Trump}) = 0$, $v_2(\text{Clinton}) = 4$, $v_2(\text{Trump}) = 6$. We can see that the outcome that maximizes social value is Clinton winning (with a social value of 14 as opposed to 6).

Now, consider the externality of each player. If player 1 weren't voting, Trump would win and player 2 would receive 6 utility; however, with player 1 voting and Clinton 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 Clark pivot rule, whereas player 2 wouldn't be charged anything.

Some things to note about this example:

- Only a "pivotal player" who can actually change the outcome of the election has to pay anything (in this case, player 1); they in addition never need to pay more than the difference in value between the two candidates.

- The same mechanism works with any number of voters and candidates.

- Why do we need to charge the voters? If we didn't charge them, they would be able to "cheat" and report false values to rig the election (what if player 2 reported his values as 0 and 100, for instance?). We return to this question in the next chapter.

Collusion. What if, in our voting example above, we have 100 voters, 98 of whom prefer Clinton, but the other two of them want to rig the election so that Trump wins and they don't have to pay anything? Let's 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 Clinton supporters. Clearly, if either one of them were to not vote, the outcome of the election would still favor Trump, so, using our VCG mechanism, *neither of them has any externality, and they would thus pay nothing!* This is an example of how collusion between players breaks the truthfulness and social optimality of the VCG mechanism. We inherently assume when reasoning about these properties that players are unable to collude and coordinate with one another in such ways.

Example: single-item auctions (recovering the second-price auction). Let us remark on how VCG works in the context of a single-item auction. 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^* was not present, the second-highest bidder i^{**} would win and receive value equal to their bid, but since i^* is present, no other player gets any value. (Note that it is important that we consider the *value* and not *utility* for players when computing the externality!) No other players are affected, and so the price p_{i^*} paid by the winner is exactly the second-highest bid. 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 Clark pivot rule gives us exactly a second-price auction when we apply it to this setting!

9.3 VCG and Matching Markets

Let us now apply VCG to the problem of designing a matching market auction.

Recall that a matching market frame $([n], Y, v)$ consists of a set $[n]$ of players, a set Y of objects, and a player valuation function $v : Y \to \mathbb{N}$. Such a matching market frame directly corresponds to a social choice 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 $i \neq j$.
- T is the set of all possible valuation functions $v : Y \to \mathbb{N}$.
- $v_i(t, a) = t(a(i))$.

We can now use the VCG mechanism with the Clark pivot rule to get a matching market auction which DST implements social value maximization (and hence social welfare maximization). Define $M(\vec{r})$ 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).

To remark on efficiency, notice that to efficiently implement this mechanism we need a way to quickly find the matching (allocation a^*) that maximizes social value. If we think of $v(t, y)$ as being the weight of edge (t, 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 7.8 gives us a polynomial-time algorithm for finding a market equilibrium, which by Corollary 7.6 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, X, T, v):

- X (the set of states) is the set of all possible allocations $a : [n] \to [n]$ such that $a(i) \neq a(j)$ if $i \neq j$.
- $T = \mathbb{N}$; 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 formulate a VCG mechanism for this context. Begin by picking the allocation a^* that maximizes social value (i.e. $\sum_{i\in[n]} c_{a^*(i)} r_j$). This amounts to assigning the k^{th} highest bid (unit price per item) to the k^{th} largest bundle (for $k = 1, \ldots, n$), breaking ties in some deterministic manner. So $a^*(i)$ is the rank of the bid r_i of player i.

Now, given the i^{th} ranked player, player $k(i)$, we must set $p_{k(i)}$ equal to that player's externality. Clearly, the optimal allocation for players ranked higher than i does not change if we remove this player, so it suffices to consider the change in value for players with rank $j > i$. If player $k(i)$ is present, these players receive total value $\sum_{j>i} c_j r_{k(j)}$. But if this player is absent, each of these players gets the next larger bundle, for total value $\sum_{j>i} c_{j-1} r_{k(j)}$. So we set the price

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

.

9.4 Generalized Second-Price (GSP) Auctions

While the VCG pricing for matching markets for bundles is relatively simple to compute, it is still a lot more complicated than the "simple" first and second price auctions for the case of single-item auction. In the case of a single-item auction, VCG actually does reduce to a second-price auction. But what happens when we try to generalize the second-price auction model to an auction with multiple items?

Specifically, let us consider a mechanism where the i^{th} ranked player $k(i)$ pays the $(i+1)^{\text{st}}$ ranked bid per item (i.e. $p_i = c_i r_{k(i+1)}$ if $i < n$ and $p_n = c_n r_{k(n)}$). This is referred to as the *generalized second-price (GSP)* mechanism. However, unlike in the single-item case, this mechanism is not DST! For example (see section 15.5/figure 15.6 of Kleinberg/Eisley), let's 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!

However, despite GSP not being truthful, it is actually the mechanism that Google and other sellers of advertising slots use for sponsored search advertisements! Among other factors, this is 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 in fact Nash-implement social value maximization:

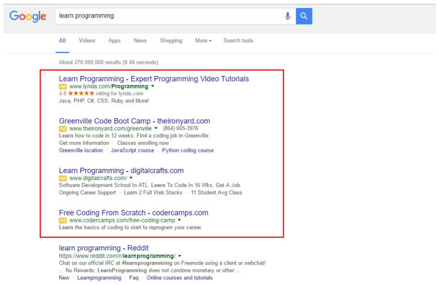
Theorem 9.10. *The GSP mechanism Nash-implements social value maximization in any market-for-bundles context.*

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} highest valuation for the items being sold.

By Theorem 7.8, there exists a market equilibrium in the matching market corresponding to this social choice context. By theorem 7.7, 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 bid and bundles by decreasing size; if a tie causes players to be assigned bundles out of order then we can reorder them accordingly).

Now let α_i be the price per item in bundle i; by Theorem 7.11 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 such pair of players, and because we can assume ties to be broken by player identity, GSP 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.



About 270,000,000 results (0.44 seconds)

Learn Programming - Expert Programming Video Tutorials
Ad www.lynda.com/**Programming**
4.8 ★★★★★ rating for lynda.com
Java, PHP, C#, CSS, Ruby and More!

Greenville Code Boot Camp - theironyard.com
Ad www.theironyard.com/greenville (864) 605-3976
Learn how to code in 12 weeks. Find a coding job in Greenville
Get more information · Classes enrolling now
Greenville location · JavaScript course · Python coding course

Learn Programming - digitalcrafts.com
Ad www.digitalcrafts.com/
Software Development School In ATL. Learn To Code In 16 Wks, Get A Job.
Ongoing Career Support · Learn 2 Full Web Stacks · 11 Student Avg Class

Free Coding From Scratch - codercamps.com
Ad www.codercamps.com/free-coding-camp
Learn the basics of coding to start to reprogram your career.

learn programming - Reddit
https://www.reddit.com/r/**learnprogramming**/
Chat on our official IRC at **#learnprogramming** on Freenode using a client or webchat!
... No Rewards: **LearnProgramming** does not condone monetary or other ...
New · Learnprogramming · Faq · Online courses and tutorials

Figure 9.3: 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.

Notice that, by construction of $\vec{r}$, player i will receive the same bundle as in the market-clearing equilibrium we assumed prior, *at the market-clearing price.*

So, if a player i deviates from its bid of r_i, they may end up with a different bundle j, and would then have to pay $r_j = \alpha_{j-1}$ per item in that bundle (a price that is at least as high as what player j is currently paying for it). But that would contradict the market-clearing property, as i would also prefer bundle j at the cheaper price α_j. ■

9.5 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 9.3. 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 a "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 advertiser 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.)

Chapter 10

Voting: Social Choice without Payments

Recall our definition of a *social choice context* (n, X, T, v) from the previous chapter:

- A set of n players.
- A finite set of states X.
- Each player i is associated with a *type* t_i in some finite *type space* T.
- Each player i is associated with a *valuation function* $v_i : T \times X \to \mathbb{N}$, where $v_i(t_i, x)$ describes how a player i having type $t_i \in T$ values a state $x \in X$.

10.1 Payment-Free Mechanisms

Let's return to the voting scenario that we had briefly considered as an example. Let's say X is a set of candidates from which we wish to elect a single leader. (Alternatively, we can think of an outcome as a set of rankings among the candidates, but for simplicity we will focus on only the task of selecting a single winner.)

As we have seen before, we can use the VCG mechanism to DST-implement social value maximization. But this required using payments, which may unfairly prioritize the votes of people with more money to spend. Democracies typically desire a voting system where everyone is treated equally—in other words, there should be no payments, and everyone's votes should be counted with equal weight.

(Remark: Even systems like the American election system are dubiously "fair" in the manner we describe here, as donations to parties or candidates may be viewed as a way to bias the electoral process.)

We refer to such a mechanism without payments (i.e. one which will always output $\vec{p} = \vec{0}$) as a *payment-free mechanism.* Unfortunately, we can show by example that the consequences of restricting the output in this manner are less than desirable:

Claim 10.1. *No payment-free mechanism can DST-implement social value maximization even in a context with only two voters and two choices ($n = |X| = 2$).*

Proof. Consider the following example with two voters (1 and 2) and two candidates (A and B):

- $t_1(A) = 2, t_1(B) = 8$
- $t_2(A) = 6, t_2(B) = 4$

Since any admissible mechanism M must output the outcome which maximizes social value, candidate B must be elected if both voters truthfully report their types. But if, say, voter 2 were to lie and report his type as $r_2(A) = 10, r_2(B) = 0$, candidate 1 would be selected as the winner, and voter 2 would increase his own utility from 4 to 6. So clearly reporting truthfully cannot be a dominant strategy in this example. ■

This proof, as we can see, applies even when each voter has a fixed "budget" of votes which it can allocate between candidates (10 in the example above), and when voters' preferences between candidates are strict.

10.2 Strategy-Proof Voting

As noted above, we can not hope to DST-implement social welfare maximization without requiring voters to pay. A reasonable first relaxation of the dominant-strategy truthfulness requirements is to only require that voters be incentivized to submit their *preference ranking* over outcomes truthfully.

Given a social choice context $\Gamma = (n, X, T, v)$, let $\text{rank}(t, i) = (x_1, x_2, \ldots, x_m)$ denote an ordering of the outcomes $x \in X$ in decreasing order according to player i's preference $v_i(x, t)$, breaking ties lexicographically.

Definition 10.2. We say that a mechanism M is a **strategy-proof voting scheme** for the context $\Gamma = (n, X, T, v)$ if for every $\vec{t} \in T^n$, we have that $\text{rank}(t, i)$ is a dominant strategy in $G^{\Gamma, \vec{t}, M}$.

That is, rational voters will always truthfully submit their preferences over candidates, and there is no situation where people will ever vote "strategically" and lie about preferences to influence the outcome (hence the name "strategy-proof").

Clearly, if we restrict our voting schemes to be strategy-proof, then we cannot hope to maximize social welfare, since there is no concept of *how much* voters prefer a particular candidate, only of the order of preference for each voter. A natural desideratum, introduced by Condorcet, would be to elect a candidate that a majority of the voters prefer to all others.

Definition 10.3. Given a set of voter preference rankings $\mu_1, \ldots, \mu_n$ over a set of candidates $X = \{x_1, \ldots, x_m\}$, 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'.

When there are only two choices, the most obvious voting scheme will suffice.

Theorem 10.4. *There exists a strategy-proof voting scheme V for all social-choice contexts over two outcomes; furthermore, V will always output a Condorcet winner.*

Proof. Let 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).[1]. By definition, the candidate x elected by 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 favored candidate; the only time when voter i's vote would matter is in the event that there would otherwise be a draw between the two candidates (or when i's vote would produce a draw instead of a loss for their candidate), and clearly in that event 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 to this case would 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 votes (this is, for instance, how American elections work at the state level), and as before, we break ties in

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

some arbitrary but fixed way. If we have two candidates, then the plurality voting rule is equivalent to the majority voting rule.

So, with more than two candidates, does plurality output a Condorcet winner? Is it strategy-proof? Perhaps surprisingly, *neither of these are true!*

Claim 10.5. *The plurality voting rule is not strategy-proof.*

Proof. Consider an example with 101 voters and three candidates (Clinton, Trump, and Johnson). Voters' preferences are as follows:

- 50 prefer $C > T > J$
- 50 prefer $T > C > J$
- 1 prefer $J > C > T$

If everyone votes truthfully, either C or T will be selected depending on how tie-breaking is implemented. Let's first assume that T is selected. If so, the player who prefers J should clearly "lie" and cast $C > J > T$ as his vote—this ensures that C wins, which is a better outcome for him. If instead ties are broken so that C wins, the player who prefers J would instead prefer to lie if his preferences had been $J > T > C$. ■

To show that plurality does not elect a Condorcet winner, we will actually show something much stronger: with three or more candidates, *no voting rule can always elect a Condorcet winner if preferences can be arbitrary!*

Theorem 10.6. *There is no mechanism V that outputs a Condorcet winner in every social choice context.*

Proof. Consider the same three candidates as above, this time with three voters and the following preferences:

1. $C > T > J$
2. $J > C > T$
3. $T > J > C$

So 2/3 of voters prefer Clinton to Trump, 2/3 of voters prefer Trump to Johnson, and 2/3 of voters prefer Johnson to Clinton. This means that there is no candidate of these three that more than half the voters prefer to each of the other two candidates, since for each candidate 2/3 of the voters prefer them to one opponent but only 1/3 to the other opponent. And so, given any mechanism V, it cannot output a Condorcet winner in this social context, because *there is no Condorcet winner in this context!* ■

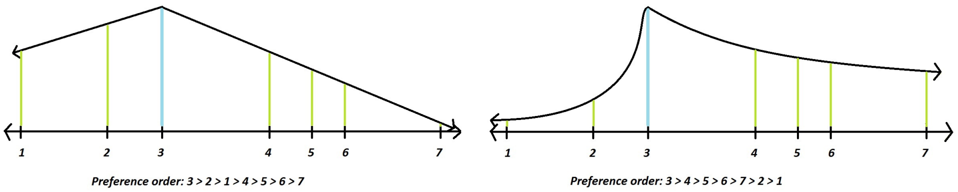


Figure 10.1: An illustration of the intuition behind single-peaked preferences. In this example, seven candidates are arranged on a spectrum. We can imagine the voter's preferences as a function over this spectrum; the only requirements for this function are that it must be maximized at the voter's top 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 "function" where $3 > 2 > 1$ and $3 > 4 > 5 > 6 > 7$ is admissible as single-peaked.

Clearly, if there is not always a Condorcet winner, we cannot hope to always elect a Condorcet winner! So a weaker alternative would be to try to elect a Condorcet winner when one exists. However, even this weaker alternative has been proven impossible by the famous GibbardSatterthwaite theorem, a result published in 1973 that further strengthens an earlier impossibility result by Kenneth Arrow's (for which Arrow won the Nobel prize in 1950). In fact, this result proves that the only voting rule over three or more candidates that is strategy-proof is dictatorship!

Theorem 10.7 (Gibbard-Satterthwaite)**.** *Let V be a strategy-proof voting rule for a social choice context (n, X, T, v) with $|X| > 2$. Then either there exists some candidate $x \in X$ that can never win (i.e. there is no profile of rankings $\vec{\mu}$ such that $V(\vec{\mu}) = x$), or V is a dictatorship (i.e. there exists a dictator i such that $V(\vec{\mu})$ is dependent on μ_i and no other voter).*

10.3 Single-Peaked Preferences

Fortunately, the above impossibility results rely on the fact that voters' preferences over candidates can be arbitrary. Under a simple natural restriction on preferences, both of them can be overcome.

We say that a social choice context Γ has **single-peaked preferences** if all outcomes 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. So 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 10.1 for an intuitive illustration.

With this notion we can prove the following elegant result.

Theorem 10.8. ***(Median Voter Theorem.)*** *There exists a strategy-proof voting rule V for all single-peaked social choice contexts; furthermore, 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, so that the median (which we will be employing in this proof) is well-defined.

Let $a_1, \ldots, a_n$ denote the top choices of each voter's reported preferences, ordered according to the one-dimensional spectrum. Let V output $a^* = a_{\frac{n+1}{2}}$ (i.e. the preference of the median voter according to the spectrum). See figure 10.2 for an example.

To show that 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 preference rule. 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.

To show that 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). So this rule is strategy-proof; no voter can strategically vote to increase their own utility.

Now, if n is even, we can just take the median vote to be $a^* = a_{n/2}$. We still have that V is strategy-proof by the same logic as above. In addition, 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^*). ■

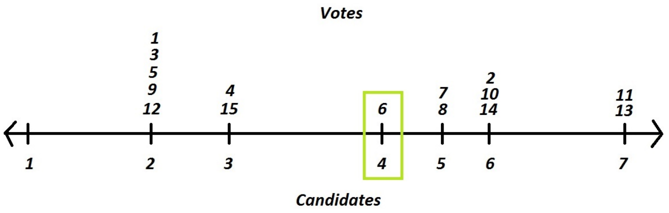


Figure 10.2: An example of the median voter rule with 15 voters and the seven candidates from the previous figure. If we order the fifteen votes according to their positions on the spectrum, voter 6 is the median (eighth) vote, and so his candidate (4) wins, despite candidate 2 having a plurality of 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!).

Notice that, in the above proof, the only difference between the cases of even and odd n is that, since we arbitrarily picked between the two possible candidates ($a_{n/2}$ and $a_{n/2+1}$) closest to the median in the even case, there may, by a symmetric argument, be another Condorcet winner ($a_{n/2+1}$) if those two candidates are distinct; in the case of an odd number of voters, we are actually guaranteed that the selected Condorcet winner is unique.

Chapter 11

Web Search

Recall that we can view the Internet as a directed graph of webpages, each node of which is a website v with directed edges (v, v') to every page v' to which it links.

Of course, each webpage also has content. The problem of web search deals with *finding the "most relevant" webpage to a given search query.* For simplicity, when studying this problem, we will focus on 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.

In fact, early web search algorithms did use such methods to score content, and as a result were very easy to fool by just creating pages with massive lists of keywords! (Most current search algorithms, in order to keep page owners from being able to artificially inflate their page's relevance score in such a way, are far more complex, as well as proprietary.)

11.1 Weighted Voting and PageRank

Given a naïve rule for scoring, can we use the graph structure of the Internet to improve upon it and create a better search algorithm?

First approach: using in-links as votes. Begin by removing all pages deemed irrelevant (by the scoring algorithm) from the graph—that is, remove all pages whose score is less than a certain threshold. Now, let a webpage's "score" under this new algorithm be the number of relevant (remaining) 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 very 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.

Weighted 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; however, in this case, a spammer can arbitrarily create new "voters"!

To circumvent this problem, we can assign a weight $w(v)$ to each webpage v and define the score of a webpage as:

$$\text{Score}(v) = \sum_{(v',v)\in E} w(v')$$

But how do we assign weights to webpages so that relevant nodes are heavily weighted and spammers' "fake" webpages aren't considered? One approach, called **Hubs-and-Authorities** (introduced by Jon Kleinberg), determines the weight of a node based on how well it is able to "predict" the score of a node. Nodes are scored based on their capability as a predictor (a *hub*) and their relevance (*authority*).

For instance, pages like Yahoo or Google News are considered hubs (as they link to different content), 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, where the weight of each node is assigned to be its 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 by many relevant pages, and so forth. Formally, we have:

$$\text{Score}(v) = \sum_{(v',v)\in E} \frac{\text{Score}(v')}{\text{out-degree}(v')}$$

And we can add the conditions that $\text{Score}(v) \geq 0$ for all v and that $\sum_v \text{Score}(v) = 1$. However, we just defined a node's score in terms of other nodes' scores, which might even be defined circularly by the score of the original node! How can we formalize this?

What we are looking for here is in fact a *fixed point* of the above equations—in other words, an assignment of scores to nodes such that all of the equations hold. In fact, we are already used to seeing and studying fixed points in game theory; the notion of a Nash Equilibrium can be viewed as a strategy that is itself a fixed point of the best-response operator for all players in a game. Let's consider a simple example of a fixed point for our PageRank algorithm:

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, so clearly

$$\text{Score}(v) = \sum_{(v',v)\in E} \frac{\text{Score}(v')}{\text{out-degree}(v')} = \frac{1/3}{1} = 1/3$$

holds for all nodes, and in addition all nodes' scores sum to 1.

We can in fact find a fixed point in a similar manner to that in which we find a PNE through iterative best-response dynamics. Consider the following iterative process:

- 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 doesn't have any out-links, all of the "flow" it carries is lost at every stage—there is a leak in the system!

Since we must ensure that the sum of all pages' scores remains 1, we henceforth will assume that nodes without out-links at least will 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 very serious issues with this procedure:

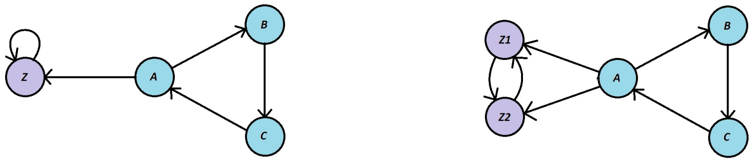


Figure 11.1: The example graphs consisting of a triangle with additional nodes. Notice, in each graph, that, if we think of PageRank score as a "flow", it can only enter the group of purple nodes and not leave; eventually, if we iterate this process, the purple nodes' total scores will converge to 1, while the others' will approach 0.

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 11.1 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 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't ever leave the network of z_1 and z_2, eventually these nodes' scores will converge to $1/2$ as 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 11.2). While the iterative process gives us a fixed point where all nodes have score $1/6$, there exist in fact 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!

11.2 Scaled PageRank

Fortunately, we can refine our iterative algorithm to deal with these problems, while also guaranteeing that it converges to a uniquely defined assignment of scores, as follows:

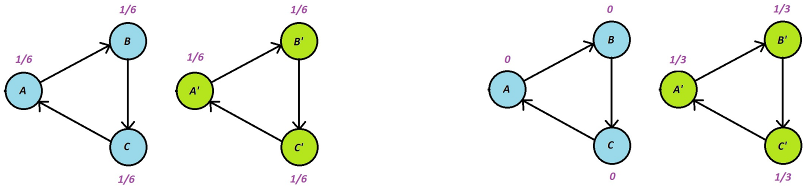


Figure 11.2: Two admissible assignments of fixed point PageRank scores for the example graph consisting of two disjoint triangles. In fact, as long as the scores of every node in a triangle are identical and the sum of all nodes' scores is 1, any such assignment is sufficient.

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, but the actual value is a closely guarded secret.)

In other words, we want a fixed point of the system:

$$\text{Score}(v) = \epsilon/n + (1-\epsilon) \sum_{(v',v)\in E} \frac{\text{Score}(v')}{\text{out-degree}(v')}$$

which we can attempt to find by modifying our iterative process:

- For each node v set $\text{Score}_0(v) = 1/n$.
- For each round i, set $\text{Score}_{i+1}(v) = \epsilon/n + (1-\epsilon)\sum_{(v',v)\in E} \frac{\text{Score}_i(v')}{\text{out-degree}(v')}$.

Now, looking at the example of the two disjoint triangles, we see that 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—for instance, if $\epsilon = 1/2$, z will get a score of $1/7$ (whereas the other nodes will have $2/7$ each).

In fact, we can show that this iterative procedure will always converge to a unique fixed point, as follows:

Theorem 11.1. *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. To show convergence, it will be immensely convenient to adopt linear-algebraic notation to represent the network. Thus:

- Give the nodes in a graph an ordering; call them $1, 2, \ldots, n$.
- 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, for our original (non-scaling) implementation of PageRank, we are looking for a score vector $\vec{r}$ such that:

$$r_i = \sum_j N_{j,i} r_j$$

or, equivalently, such that $\vec{r} = N^T \vec{r}$ (N^T being the transpose of N).

Hence, what we are looking for is an *eigenvector* of N^T with *eigenvalue 1!* (Recall from linear algebra that an eigenvector $\vec{v}$ with eigenvalue λ to a matrix M is defined as a vector satisfying $M\vec{v} = \lambda\vec{v}$.)

Unfortunately, we must also satisfy the additional constraints that all entries of $\vec{r}$ are non-negative reals and that the L_1-norm of $\vec{r}$, $||\vec{r}||_1 = \sum_i |r_i|$, is equal to 1. Does such an eigenvector always exist?

In fact, a powerful theorem from linear algebra, originally proposed by Oskar Perron in 1907, shows that an eigenvector $\vec{v}$ with eigenvalue and L_1-norm 1 and non-negative real-valued entries will always uniquely exist for any matrix M subject to the following conditions:

- Every entry in M is a *strictly positive* real number.
- For any vector $\vec{x}$ with $||\vec{x}||_1 = 1$, $||M\vec{x}||_1 = 1$.

While this second condition may seem daunting, note the fact that multiplying by the matrix N^T corresponds to applying a single phase of our 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.

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:

$$r_i = \epsilon/n + (1 - \epsilon) \sum_j N_{j,i} r_j$$

But $||\vec{r}||_1 = 1$, so, equivalently:

$$r_i = \sum_j ((1-\epsilon)N_{j,i} + \epsilon/n)r_j$$

So, let us define a new matrix $\tilde{N}$ such that $\tilde{N}_{i,j} = (1-\epsilon)N_{i,j} + \epsilon/n$. Now we see that $r_i = \sum_j \tilde{N}_{j,i} r_j$, 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 $\vec{r}$ that satisfies this. ■

Next, we show that the iterative procedure for scaled PageRank not only converges, but does so quickly.

Theorem 11.2. *The iterative procedure described above for scaled PageRank converges to the scores uniquely defined above.*

Proof. It will be of great use to us to define our iterative procedure in terms of the linear-algebraic notation we used in the previous proof, as follows:

- $\vec{r}^0 = 1/n(\vec{1})$
- $\vec{r}^{i+1} = \tilde{N}^T \vec{r}^i = (\tilde{N}^T)^i \vec{r}^0$

Additionally, 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:

$$\begin{aligned}
\text{Err}(t+1) = ||\vec{r^{t+1}} - \vec{r^*}||_1 &= \\
||\tilde{N}^T \vec{r^t} - \tilde{N}^T \vec{r^*}||_1 &= \\
||((1-\epsilon)N^T \vec{r^t} + \frac{\epsilon}{n}(\sum \vec{r_i^t})\vec{1}) - ((1-\epsilon)N^T \vec{r^*} + \frac{\epsilon}{n}(\sum \vec{r_i^t})\vec{1})||_1 &\leq \\
||((1-\epsilon)N^T \vec{r^t} + \frac{\epsilon}{n}||\vec{r_i^t}||_1\vec{1}) - ((1-\epsilon)N^T \vec{r^*} + \frac{\epsilon}{n}||\vec{r_i^t}||_1\vec{1})||_1 &= \\
||((1-\epsilon)N^T \vec{r^t} + \frac{\epsilon}{n}\vec{1}) - ((1-\epsilon)N^T \vec{r^*} + \frac{\epsilon}{n}\vec{1})||_1 &= \\
(1-\epsilon)||N^T \vec{r^t} - N^T \vec{r^*}||_1 &= \\
(1-\epsilon)||N^T(\vec{r^t} - \vec{r^*})||_1 &= \\
(1-\epsilon)||(\vec{r^t} - \vec{r^*})||_1 &= \\
(1-\epsilon)\text{Err}(t)&
\end{aligned}$$

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!). The probability that you end up at a certain page after k steps is, in fact, given by the distribution of unscaled PageRank scores:

- After 0 steps: $\vec{r^0}$ (uniform distribution)
- After 1 step: $N^T \vec{r}_0 = \vec{r^1}$
- After 2 steps: $N^T \vec{r}_1 = \vec{r^2}$
- and so on...

So the k-iteration PageRank 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 as 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 random node, and each time we are randomly transported (with probability ϵ) we end up at a uniformly random node. In an alternative version, we can attempt to obtain a personalized score for a certain type of user by changing the distribution from uniformly random to something different—for instance, how popular each page is (or is estimated to be) among the user's demographics.

11.3 Impossibility of Non-manipulable Websearch

Search-engine optimization...**To be completed**

Chapter 12

Beliefs: The Wisdom and Foolishness of Crowds

12.1 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 *average* of the crowd's estimates tends to be a fairly good estimate. A famous example of this was a crowd of people attempting to guess the weight of an ox at a county fair; most people were individually far off, but when the average of everyone's guesses was taken, it was surprisingly close!

Let us introduce a simple model to explain this phenomenon. We will focus on a simple "yes/no" 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's assume we have a set of n individuals. None of them have any preference between "yes" or "no" beforehand, but then each individual i *independently* receives some signal X_i that is correlated with the state of the world, but only very weakly so. For instance, given a small $\epsilon > 0$, we might have

$$\Pr[X_i = W] \geq 1/2 + \epsilon$$

where all X_i are independent random variables.

Each individual will then simply report their signal X_i (so, if they saw evidence that $W = 0$, they report 0; if they saw evidence that $W = 1$, they report 1). This would be the case, for instance, if each individual were to receive a high utility for guessing correctly and a low utility for guessing incorrectly. (We can formally model this through the notion of a Bayesian game, but we will not do so for simplicity's sake.) Note, however, that we assume that these guesses happen *simultaneously*, so that a player can't be influenced by others' guesses, only by their own evidence.

What is the probability that the "average" of these guesses—that is, the "majority bit" b that got the most guesses—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 uninformed and only has that probability of guessing correctly, so why would majority voting increase the chances of producing an "informed" result?

Indeed, this would be the case if the signals were *dependent*, and players had all received the same signal. However, we are considering a world where each player receives a "fresh", *independent* signal that has nothing to do with what the other players have observed. In this case, we can use a variant of the Law of Large Numbers to argue that, with very high probability (dependent on n), the majority vote will in fact equal W! To that end, we will introduce the Chernoff-Hoeffding bound, which is a quantitative version of the Law of Large Numbers:

Theorem 12.1 (Chernoff-Hoeffding Bound)**.** *Let $Y_1, \ldots, Y_n$ be n independent random variables such that $|Y_i| \leq 1$. Let $M = \frac{1}{n}\sum_{i=1}^{n} Y_i$. Then:*

$$Pr[|M - \mathbb{E}[M]| \geq \epsilon] \leq e^{-\frac{1}{2}\epsilon^2 n}$$

In other words, the Chernoff-Hoeffding bound is a bound on the deviation of the "empirical mean" of independent random variables from the expectation of the empirical mean.

We will omit the proof of this theorem; instead, we will prove that in the experiment above the majority vote will be correct with high probability. Let Majority$(x_1, \ldots, x_n) = 0$ if at least $n/2$ of the x_i are zero, and 1 otherwise. Then we have:

Theorem 12.2. *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:*

$$Pr[Majority(x_1, \ldots, x_n) = W] \geq 1 - e^{-2\epsilon^2 n}$$

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 Majority$(X_1, \ldots, X_n) = W$ if and only if $M = \frac{1}{n}\sum_{i=1}^{n} Y_i > 0$. We will show that $\Pr[M \leq 0]$ is sufficiently small, which by the above implies the theorem.

By linearity of expectation, we have $\mathbb{E}[M] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[Y_i]$. Note that, for all i, $\mathbb{E}[Y_i] \geq 1(\frac{1}{2}+\epsilon) + (-1)(\frac{1}{2}-\epsilon) = \frac{1}{2}+\epsilon-\frac{1}{2}+\epsilon = 2\epsilon$.

So $\Pr[M \leq 0] \leq \Pr[|M - 2\epsilon| \geq 2\epsilon] = \Pr[|M - \mathbb{E}[M]| \geq 2\epsilon]$, which by the Chernoff-Hoeffding bound is smaller than $e^{-\frac{1}{2}(2\epsilon)^2 n} = e^{-2\epsilon^2 n}$. ■

Observe that, as we might expect, the greater the number of players n and the greater the certainty ϵ that players' guesses are correct, the higher the lower bound on the probability that the average guess is correct.

Connection to two-candidate voting. The above theorem shows that the majority voting rule (studied in the previous chapter) has the property that, if the outcomes over which players vote are objectively "good" or "bad" (but players are uncertain about which is which), majority voting will lead to the "good" outcome as long as players receive independent signals sufficiently correlated with the correct state of the world.

12.2 The Foolishness of Crowds

Let's now change the problem a bit. Instead of having the players announce their guesses at the same time, instead have players announce them *sequentially*, where, before making their own guess, a player is allowed to first observe the guesses of everyone who guessed before them. (For instance, think of this as posting your opinion on Facebook after first seeing what some of your friends have posted.)

How should people act in order to maximize the probability that their own guess is correct? (Again, we can think of a high utility being awarded for a correct guess and a low utility for an incorrect guess.)

For simplicity, let us assume that all players' evidence is equally strong—that is, $\Pr[X_i = W] = 1/2 + \epsilon$ for all i. Additionally, recall our earlier assumption that players are indifferent a priori (they believe initially that $\Pr[W = 1] = 1/2$).

So, what will the first player do? Intuitively, they will act exactly as before, and guess whatever their own evidence tells them (guess $g_1 = X_1$), since they have no additional information.

The second player, however, effectively has two pieces of evidence now—their own and also the first player's. Intuitively, if $g_1 = 1$ and $X_2 = 1$, they should guess $g_2 = 1$, since they now have two pieces of evidence favoring $W = 1$.

To formalize this notion, we will rely on:

Theorem 12.3 (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:

$$\Pr[B \mid A]\Pr[A] = \Pr[A \cap B] = \Pr[A \mid B]\Pr[B] \qquad \blacksquare$$

So the second player should guess $g_2 = 1$ after observing $g_1 = X_2 = 1$ if:

$$\Pr[W = 1|X_1 = X_2 = 1] \geq \Pr[W = 0|X_1 = X_2 = 1]$$

By Bayes' Rule,

$$\Pr[W = 1|X_1 = X_2 = 1] = \Pr[X_1 = X_2 = 1|W = 1]\frac{\Pr[W = 1]}{\Pr[X_1 = X_2 = 1]}$$

$$= \frac{(\frac{1}{2} + \epsilon)^2}{2\Pr[X_1 = X_2 = 1]}$$

By similar logic,

$$\Pr[W = 0|X_1 = X_2 = 1] = \frac{(\frac{1}{2} - \epsilon)^2}{2\Pr[X_1 = X_2 = 1]}$$

which is clearly smaller.

So, if the second player sees either $g_1 = X_2 = 1$ or $g_1 = X_2 = 0$, they should output that guess. If $g_1 = 1$ but $X_2 = 0$, or vice versa, then intuitively $W = 0$ and $W = 1$ are equally likely (which can again 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.

But what about the third player? Specifically, what happens if the third player sees, for instance, $g_1 = g_2 = 1$? In this case, *no matter what X_3 is, the third player will be better off guessing 1!* 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.

Of course, if this should happen, then the fourth player, even knowing that the third player ignored their evidence, would also ignore their own and output $g_4 = g_2 = g_1$ as well. And so we end up with a *cascade*, where everyone votes in accordance with the first two players and ignores their evidence. (In general, this will occur at any point where the number of guesses for either 0 or 1 outnumber the other by 2!)

Notice that, if, say, $W = 0$, a cascade where *everyone guesses the incorrect state* $W = 1$ happens as long as $X_1 = X_2 = 1$, which occurs with probability $(\frac{1}{2} - \epsilon)^2$. Even with a relatively large ϵ—say, 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 *everyone* will guess incorrectly! 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.

There are some unrealistic aspects of this model, however. 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 rational agents weight every piece of evidence equally. Furthermore, this 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.

12.3 More on Belief Updates

To determine whether a cascade occurred, we did not have to fully determine the beliefs of the players. But in other circumstances that is necessary. We consider two simple examples that arise commonly in practice: a) how to interpret the result of a medical test, and b) how to determine whether an email is spam, based on the occurance of certain keywords.

Interpreting the Results of a Medical Test

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 population tested positive, what is the probability that he has the disease? The answer is not 0.99! To answer this question, we consider a slight variant of Bayes' rule which relies on the following simple property of conditional probabilities:

Claim 12.4. *Let $A_1, \dots, A_n$ be disjoint events with non-zero probability such that $\bigcup_i A_i = S$ where S is the sample space (i.e., the events are an exhaustive partitionning of 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 $A_1, \dots, A_n$ are disjoint it follows that the events $B \cap A_1, \dots, B \cap A_n$ are also disjoint. Therefore

$$\sum_{i=1}^{n} \Pr[B \cap A_i] = \Pr\left[\bigcup_{i=1}^{n} B \cap A_i\right] = \Pr\left[B \cap \bigcup_{i=1}^{n} A_i\right] = \Pr\left[B \cap S\right] = \Pr[B]$$

■

By combing Bayes' rule with this claim, we get the following useful variant of Bayes' rules:

Theorem 12.5 (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 12.4, 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's 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 population, 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 12.6 (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):

$$\Pr\left[\text{spam} \mid \bigcap_i W_i\right] = \frac{\prod_i \Pr\left[W_i \mid \text{spam}\right]}{\prod_i \Pr\left[W_i \mid \text{spam}\right] + \prod_i \Pr\left[W_i \mid \text{not spam}\right]}$$

Chapter 13

Knowledge and Common Knowledge

13.1 The Muddy Children Puzzle

Suppose a group of children are in a room, and some of them have mud on their foreheads. All of the children can see any other child's forehead, but are unable to see, feel, or otherwise detect whether they have mud on their own.

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, in particular, 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 13.1 (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. 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 true if the claim is true for n children. 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.

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, the induction hypothesis provides that 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.

Of course, if there are fewer than $n+1$ muddy children, the induction hypothesis provides that they all will have said "yes" before round $n+1$. This completes the induction. ■

13.2 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, and independently but subsequently by the economist Robert Aumann (who received the Nobel Prize in 2005). We will begin with a treatment that more closely follows Kripke's work, but proceed to applications similar to those studied by Aumann.

Assume we have a set of possible "worlds" Ω. Each such possible world $\omega \in \Omega$ specifies some "outcome" $s(\omega)$—for instance, in the muddy children example, s specifies which children are muddy; in a single-item auction, s might specify the valuations of each player for the item. Additionally, each world specifies "beliefs" for all of the players; $B_i(\omega)$ is the set of all worlds that a player i considers to be possible at ω. So, roughly speaking, we can say that a player i "knows" some statement ϕ if ϕ is true in every world i considers to be possible. We can formalize this concept as follows:

Definition 13.2. A (finite) knowledge structure tuple $M = ([n], \Omega, X, s, \vec{B})$ is 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 $i \in [n]$, $B_i : \Omega \to 2^\Omega$ maps worlds to sets of possible worlds; we refer to $B_i(\omega)$ as the beliefs of i at ω.
- For all $\omega \in \Omega$, $i \in [n]$, it must hold that $\omega \in B_i(\omega)$. (Players must believe that the true state of the world is possible!)
- For all $\omega \in \Omega$, $i \in [n]$, and $\omega' \in B_i(\omega)$, it must hold that $B_i(\omega') = B_i(\omega)$. (Players can't consider it possible that they would ever believe something that they don't currently believe.)

Returning to those last two conditions, the second-to-last implies that players 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 clearly possible. (This condition is, however, not always appropriate, since people may sometimes be fully convinced of false statements! We will return to this later when discussing the difference between knowledge and beliefs.)

The last condition, meanwhile, implies that players "know" their beliefs—that, in every world a player considers possible at ω, their beliefs will be the same as at ω.

We typically represent this structure as a "knowledge network". Nodes are states of the world, and are labeled with the outcome in this world. Edges, labeled by player names i, exist between states ω and ω' whenever $\omega' \in B_i(\omega)$ (they correspond to the belief operator). Notice that the second-to-last condition implies that every node will have a self-loop.

Example. Consider a single-item auction with two buyers. Let the outcome X be a pair consisting of the players' valuations for the item. Assume there are two possible worlds, ω_1 and ω_2, such that $s(\omega_1) = (10, 5)$ and $s(\omega_2) = (10, 3)$. In addition, $B_1(\omega_1) = B_1(\omega_2) = \{\omega_1, \omega_2\}$, $B_2(\omega_1) = \omega_1$, and $B_2(\omega_2) = \omega_2$. The knowledge network graph for this example is given in Figure 13.1.

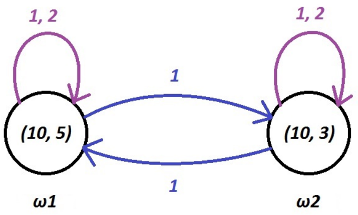


Figure 13.1: A knowledge network for a single-item auction, where the world state is dependent on player 2's valuation. This captures the fact that player 2 is fully aware of his valuation, but player 1 is not.

So, in either world, player 1 knows his own valuation (10), but he does not know player 2's valuation, only that it can be either 3 or 5. In contrast, player 2 knows both his own and player 1's valuation for the item in either world.

If we were to change the structure so that $B_1(\omega_1) = \{\omega_1, \omega_2\}$ still, but $B_1(\omega_2) = \{\omega_2\}$, this would no longer be a valid knowledge structure, by the last condition (since $\omega_2 \in B_1(\omega_1)$ but $B_1(\omega_1) \neq B_1(\omega_2)$); in particular, player 1 would consider it possible that he knows player 2's valuation is 3 in a state ω_1 where he does not actually know it, which is clearly inconsistent.

Note that a world determines not only players' beliefs over outcomes (valuations, in the above example), but 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*.

Now, let us return to the muddy children example using this formalism; assume for simplicity that we have two children. Our space X of possible outcomes is $\{M, C\}^2$ (whether each child is muddy or clean). We can define four possible worlds $\omega_1, \dots, \omega_4$, where:

- $s(\omega_1) = (M, M)$
- $s(\omega_2) = (M, C)$
- $s(\omega_3) = (C, M)$
- $s(\omega_4) = (C, C)$

By the rules of the game, we have some restrictions on each player's beliefs:

- $B_1(\omega_1) = B_1(\omega_3) = \{\omega_1, \omega_3\}$ (player 1 knows player 2 is muddy, but cannot tell whether or not he himself is.)

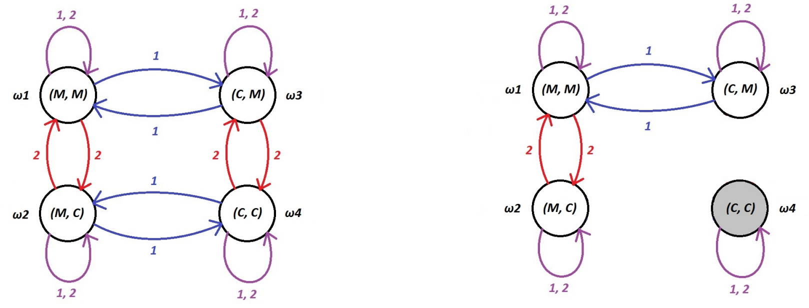


Figure 13.2: Left: the knowledge network for the muddy children example. Right: The knowledge network when the father announces that at least one child is muddy. Notice that, unless the state is ω_1, one of the two children is now able to determine the current state.

- $B_1(\omega_2) = B_1(\omega_4) = \{\omega_2, \omega_4\}$ (similarly, if player 2 is clean; note that we assume that the children don't know that some children are muddy until the father tells them.) Similarly,
- $B_2(\omega_1) = B_2(\omega_2) = \{\omega_1, \omega_2\}$, and
- $B_2(\omega_3) = B_2(\omega_4) = \{\omega_3, \omega_4\}$

See Figure 13.2 for the corresponding knowledge network. To reason about this game, let us formally define what it means for a player i to know that some event ϕ holds. Recall that, in our definition so far, we said that i knows ϕ if ϕ holds in every world i considers possible.

First, let us define an event: An *event* ϕ is a subset of states $\omega \in \Omega$. For instance, we can define MUDDY_i as the set of worlds where player i is muddy, and $\mathsf{SOMEONE\_MUDDY} = \bigcup_i \mathsf{MUDDY}_i$. Naturally, an event ϕ holds in a world ω if $\omega \in \phi$.

Now we can define the event $\mathbf{K}_i\phi$ ("player i knows ϕ") as the set of states ω where "i knows ϕ"–that is, ϕ is true in all worlds i considers possible at ω. Formally:

Definition 13.3. Given a knowledge structure tuple $M = ([n], \Omega, X, s, \vec{B})$ and an event $\phi \subset \omega$, let $\mathbf{K}_i^M\phi$ denote the subset of worlds ω where $B_i(\omega) \subset \phi$.

Whenever M is clear from the context, we can simply write $\mathbf{K}_i\phi$. In addition, we can define $\mathbf{E}\phi = \bigcap_i \mathbf{K}_i\phi$ to be the set of worlds where *everyone* knows ϕ.

Now, returning once more to the muddy children example, let's assume we have two children, both of whom are muddy. So the true state ω of the world is ω_1 as defined above.

Notice that $\omega \in \mathbf{K}_1\,\mathsf{SOMEONE\_MUDDY}$, since player 1 knows that someone (player 2) is muddy. Similarly, $\omega \in \mathbf{K}_2\,\mathsf{SOMEONE\_MUDDY}$, and so $\omega \in \mathbf{E}\,\mathsf{SOMEONE\_MUDDY}$.

Furthermore, $\omega \in \neg\mathbf{K}_i\,\mathsf{MUDDY}_i$ (denoting by $\neg\phi$ the complement of the event ϕ), since nobody initially knows whether they are muddy or not.

Does the father's announcement that there is some muddy child tell the children something that they do not already know? It might seem at first that it does not, since each child already knows that the other is muddy. However, since the announcement was public, something actually does change; we can reason about the knowledge network to see what this is.

Clearly, everyone now knows that the state ω_4 is not possible. So this changes the knowledge graph, as seen in the right of Figure 13.2.

In particular, before the announcement, everyone knew that ω_4 was impossible. However, now, not only does everyone know that, everyone also knows that *everyone knows that* ω_4 is impossible. So, while previously player 1 might have considered it possible that the current state is ω_3, he himself is clean, and so player 2 considers it possible that he too is clean and the current state is ω_4, *this is no longer possible!*

Formally, we can say that $\omega \in \mathbf{E}\ \mathbf{E}\ \mathsf{SOMEONE\_MUDDY}$. Additionally, we also have $\omega \in \mathbf{E}\,\mathbf{E}\ \ldots\ \mathbf{E}\,\mathsf{SOMEONE\_MUDDY}$ ("everyone knows that everyone knows that... everyone knows that someone is muddy"); we say then that it is *commonly known* that someone is muddy, and can define it thus:

Definition 13.4. Let $\mathbf{C}\,\phi$ (the common knowledge of ϕ) be the event $\bigcap_i \mathbf{E}_i\,\phi$.

So $\omega \in \mathbf{C}\,\mathsf{SOMEONE\_MUDDY}$. The important takeaway here is that when an announcement is made, not only do we learn the fact of the announcement, but it becomes common knowledge. As we shall see, this difference has an important impact.

Returning to the puzzle, does the announcement of the father enable anyone to know whether they are muddy? No, in fact; even in the graph without ω_4, we still have $\omega \in \neg\mathbf{K}_i\,\mathsf{MUDDY}_i$. So everyone will reply "no" to the first question.

How does this announcement change the players' knowledge? In fact, 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 have complete knowledge 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.

Justified True Belief and the Gettier problems. In our treatment, we are assuming the knowledge structure is exogenously given (for instance, in the muddy children example, it was explicitly given in the problem description). Sometimes, however, it is not even clear how the right knowledge structure for a situation should be defined. The classic paradigm in philosophy was to define knowledge as "justified true belief"—if you believe ϕ, and have some reason to believe it, then ϕ is true.

However, there are several issues to this approach, as demonstrated by the now-famous "Gettier problems". For instance, let's say you walk into a room, you observe that the thermometer says it is 70 degrees Fahrenheit, and consequently you believe the temperature is 70 degrees. According to JTB, you know that the temperature is 70 degrees, since you believe it and have a reason for doing so.

But what if, instead, the reason that the thermometer says 70 degrees is because it is broken or stuck? Do you still know that it is 70 degrees in the room? Most people would argue that you would not know, but according to JTB you would know.

13.3 Knowledge vs. Belief: Explaining Bubbles in the Market

So far in our discussion, we have described 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—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 it 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).

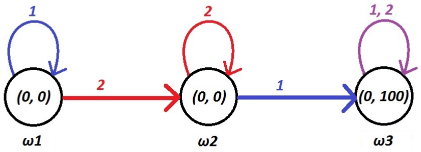


Figure 13.3: A belief network for a second-price auction. Some states in a belief network (here, ω_1 and ω_2) lead to players having incorrect beliefs.

To see how this makes things different, let us once more consider an auction scenario with two players and a single item; see Figure 13.3.

Assume that the state of the world is ω_1. Player 1 knows the true state of the world. However, player 2 has an incorrect belief, as 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 actually values it at zero!

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 does everyone believe this item is worthless, everyone believes that everyone thinks the item is worthless!

However, 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 knowledge" in a belief context that everyone is rational).

In essence, this will produce a "bubble" in the market. We will briefly and informally explain how this can happen; a formal paper can be found in a recent econometrics paper (Chen, Micali, Pass; 2016).

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! This process works as follows:

- If the true state were actually ω_3, player 2 should bid 100, since bidding less would only result in potentially losing the object.
- So, in ω_2, since player 1 believes (incorrectly) that the correct state is ω_3, player 1 should bid 99, so that he will lose and not have to pay anything while collecting as much of the participation reward as possible.
- Thus, in the actual state ω_1, player 2, who believes incorrectly that the current state is ω_2, will bid 98 (under the assumption that player 1 believes the state to be ω_3 and bids 99), in order to lose the auction while collecting as much of the participation reward as possible.
- But player 1, in state ω_1, will then bid 97 under the assumption (this time correct) that player 2 believes the state to be ω_2.

And so, in the actual world (ω_1), player 1 will bid 97 and player 2 will bid 98; thus, the item will be sold to player 2 for 97, and the auctioneer receives a utility of 97 (minus the reward R) for a worthless item!

13.4 Knowledge and Games

Now we shall use our model of knowledge to reason about games. We must first define what it means for a knowledge structure to be appropriate for a game G:

Definition 13.5. We say that a knowledge structure tuple $M = (n, \Omega, X, s, \vec{B})$ 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 $\omega' \in B_i(\omega)$, $s_i(\omega') = s_i(\omega)$, where s_i denotes the i^{th} component of s. (Players always know their own strategy.)

In other words, each world models what action each of the players takes; players have beliefs about what others are doing and believing, and we assume players always know what they themselves will do.

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

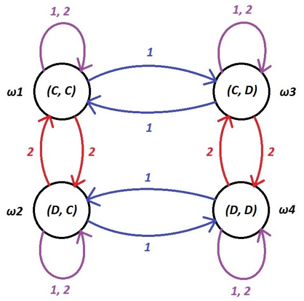


Figure 13.4: A knowledge network for the Prisoner's Dilemma. This graph should look familiar, but notice that unlike the muddy children example, where each player knows the other player's state and not their own, here each player knows their own strategy and not the other's. Also observe that, while there is one state per action profile here, this is not necessarily always true; other networks might have several nodes with the same action profile but different beliefs.

be rational; basically, we define what it means to not be "crazy" or "totally 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 isn't strictly better off switching to a'_i. In other words, if player i is acting rationally, there exists no action a'_i such that $s_i(\omega)$ is strictly dominated by a'_i in every world i considers possible.

$$\mathsf{RAT}_i = \{\omega \in \Omega \mid \forall a'_i \; \exists \; \omega' \in B_i(\omega) : u_i(a'_i, s_{-i}(\omega')) \leq u_i(s(\omega'))\}$$

$$\mathsf{RAT} = \bigcap_{i \in [n]} \mathsf{RAT}_i$$

(the event that everyone is "rational").

Based on this observation, we can get the following simple characterizations of states where RAT holds:

Theorem 13.6. *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 a round of iterative strict dominance).*

2. *There exists a knowledge structure $M = (n, \Omega, X, s, \vec{B})$ that is appropriate for G and a state $\omega \in \Omega$ such that $s(\omega) = \vec{a}$ and $\omega \in \mathsf{RAT}$.*

Proof. **(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 $\vec{a}$. In addition:

- Let $s(\omega_{\vec{a}}) = \vec{a}$.
- Let $B_i(\omega_{\vec{a}}) = \{(a_i, a_{-i}) | a_{-i} \in A_{-i}\}$ (that is, i has no idea what any other player is doing, but knows their 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 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 $\omega \in \mathsf{RAT}$.

Now assume for the sake of 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 $\omega' \in B_i(\omega)$ such that

$$u_i(a'_i, s_{-i}(\omega')) \leq u_i(a_i, s(\omega'))$$

But, because players know their own strategy, $s_i(\omega') = a_i$. Thus,

$$u_i(a'_i, s_{-i}(\omega')) \leq u_i(a_i, s_{-i}(\omega'))$$

which contradicts the fact that a'_i strictly dominates a_i. ■

It seems natural to assume not only that players are themselves rational, but also that they know that players are rational, that they know that they know that players are rational, and so forth. In other words, what happens if we assume *common knowledge of rationality* (CKR)?

The following theorem, closely related to the last, shows that CKR characterizes the set of strategies that survive ISD.

Theorem 13.7. *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 $M = (n, \Omega, X, s, \vec{B})$ that is appropriate for G and a state $\omega \in \Omega$ such that $s(\omega) = \vec{a}$ and $\omega \in$ **C** RAT.*

Proof. **(1) → (2).** Let ISD be the set of strategies surviving ISD. Construct a knowledge structure M where we have a state $\omega_{\vec{a}}$ for every possible action profile $\vec{a}$. In addition:

- Let $s(\omega_{\vec{a}}) = \vec{a}$.
- Let $B_i(\omega_{\vec{a}}) = \{(a_i, a_{-i}) | a_{-i} \in ISD_{-i}\}$ (that is, i has no idea what any other player is doing, but knows their own action and that other players will play an action profile that survives ISD).

Just as before, we can easily verify that in every world players know their beliefs and their strategy. We additionally have the following claim:

Claim 13.8. *$\omega \in$ RAT for every $\omega \in \Omega$.*

Proof. Assume for the sake of contradiction that there exists some state ω and some player i where $\omega \notin \mathsf{RAT}_i$. That is, there exists some a'_i that strictly dominates $a_i = s_i(\omega')$ with respect to $s_{-i}(\omega')$ for all $\omega' \in B_i(\omega)$.

By our construction of M (specifically, B), a'_i strictly dominates a_i with respect to ISD_{-i} and thus should be deleted by ISD! The only reason this could possibly not be true 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^1_i; in turn, this strategy is either in ISD_i or was removed due to being dominated by a^2_i. Inductively, we eventually will reach $a^m_i \in ISD_i$, which will strictly dominate 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 $a_i \in ISD_i$. ■

Now that we have this claim, it follows that, for each $\omega \in \Omega$, $\omega \in \mathbf{K}_i$ RAT; thus, $\omega \in \mathbf{E}$ RAT, $\omega \in \mathbf{E}\ \mathbf{E}$ RAT, and so on. We inductively have $\omega \in \mathbf{C}$ RAT.

By the above and our construction of M, we have the necessary conditions for every strategy profile $\vec{b} \in ISD$ and its corresponding state $\omega_{\vec{b}}$.

(2) → (1). Consider some knowledge structure M appropriate for G. Let ISD^k denote the set of strategies surviving k rounds of ISD. We shall prove by induction that for any state $\omega \in \mathbf{E}^k$ RAT, we have $s(\omega) \in ISD^k$. The base case ($k = 0$) is proven by Theorem 13.7 above.

For the inductive step, assume the statement is true for k, and let us prove it for $k+1$. Consider an $\omega \in \mathbf{E}^{k+1}$ 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, $\omega \in B_i(\omega)$; consequently, $\omega \in \mathsf{RAT}$ as well.

So, by the induction hypothesis, $s_i(\omega) \in ISD_i^k$; furthermore, for every $\omega' \in B_i(\omega)$, $s_{-i}(\omega') \in ISD_{-i}^k$.

Since $\omega \in \mathsf{RAT}$, 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 $s_i(\omega) \in ISD_i^{k+1}$ (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 $\omega \in \mathbf{C}\,\mathsf{RAT}$, $s(\omega) \in ISD^k$ for any k, implying that $s(\omega)$ survives ISD. ■

Can we hope to also produce a characterization of PNEs? Here, we retain the rationality condition, but also add a condition that everyone knows the actions played by everyone else, in addition to their own. Specifically, let KS be the event that all players know every player's strategy; formally,

$$\mathsf{KS} = \{\omega \in \Omega \mid \forall i \in [n], \omega' \in B_i(\omega) : s(\omega') = s(\omega)\}$$

Theorem 13.9. *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 $M = (n, \Omega, X, s, \vec{B})$ that is appropriate for G and a state $\omega \in \Omega$ such that $s(\omega) = \vec{a}$ and $\omega \in \mathsf{KS} \cap \mathbf{C}$ RAT.*
3. *There exists a knowledge structure $M = (n, \Omega, X, s, \vec{B})$ 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. **(1) → (2).** Consider a structure M with just one state ω such that $s(\omega) = \vec{a}$ and $B_i(\omega) = \omega$. Clearly, at ω, KS holds and $\mathbf{C}$ RAT holds by trivial induction.

(2) → (3). Trivial.

(3) → (1). KS and RAT taken together imply that each player i is playing a best response $s_i(\omega)$, and so $s(\omega)$ is a PNE. ■

Stronger notions of rationality and probabilistic structures. So far we have considered only a very weak version of rationality—basically, that a player is rational implies only that they are not acting totally unreasonably in some world that they believe possible. We can also consider stronger notions of rationality. For instance, a natural first step would be to require for $\omega \in \mathsf{RAT}_i$ that i consider some world ω' possible where $s_i(\omega)$ is a best response (that is, there exists $\omega' \in B_i(\omega)$ such that $s_i(\omega) \in BR_i(s(\omega)))$. The characterization of a PNE remains the same, but common knowledge of rationality now is characterized an alternative deletion process that can be thought of as "iterative removal of never-best-responses" where at each step we remove all actions that can never be a best response.

An even 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. We can now require that each player plays a best response to their own beliefs to be considered rational—that is, that a_i maximizes expected utility given the distribution of a_{-i} induced by their beliefs. CKR still is characterized by ISD in this setting as long as we change the definition of ISD to also allow domination by "mixed strategies" (that is, we remove actions that are strictly dominated by a probability distribution over other actions). Analogous characterizations of (mixed-strategy) Nash equilibria may also be obtained, but we omit the details here.

Chapter 14

Markets with Network Effect

Recall our previous model of networked coordination games, where each action had an intrinsic value to a player and some coordination ("network") value depending on how many of their neighbors chose it.

We shall 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 a good for sale, but buyers will buy at most one.

14.1 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 this 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.

First, think about a simple market with just an *intrinsic value* for the good, but assume players can have different intrinsic values (formally, these values can be modeled as players' types). We assume we have a large set of buyers; let $F(p)$ denote the fraction of the buyers whose intrinsic value t for the good is at least p. Thus, in the absence of network effect, if the good is priced at p, then $F(p)$ of the buyers will buy it.

Now, we can add network effect and coordination to this. We say that the value of a good for a player with intrinsic value t is

$$v = td(x)$$

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 will consider $d(x)$ monotonically increasing, so that the more users have the good, the more valuable it is; however, it might 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, "Nobody goes there anymore. It's too crowded.")

So, 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, the buyer will agree to buy it if $t \geq \frac{p}{d(x)}$.
- So, if *everyone* believes that x fraction of the population will buy the good, then an $F\left(\frac{p}{d(x)}\right)$ fraction of buyers will actually buy it.

Of course, beliefs may be incorrect. So, if people initially believe that, say, a $x_0 = 0$ fraction of the players will buy the good, then $x_1 = F\left(\frac{p}{d(x_0)}\right)$ will buy it; but, given this, $x_2 = F\left(\frac{p}{d(x_1)}\right)$ should actually buy it, and so on.

Consequently, following the general theme of this course (and analogously to some of the earlier discussions about topics such as BRD and PageRank), we are interested in the "stable points" or equilibria of this system, where buyers' beliefs are correct. We refer to these as *self-fulfilling equilibria*—that is, situations where if everyone believes that an x fraction will buy the good, then an x fraction will actually buy it.

More precisely, we refer to a belief x as a self-fulfilling equilibrium if $F\left(\frac{p}{d(x)}\right) = x$.

Example. Consider $F(p) = \max(1-p, 0)$. (So nobody has intrinsic value higher than 1, but everyone has at least 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).

Then self-fulfilling beliefs are characterized by

$$x = F\left(\frac{p}{d(x)}\right) = \max\left(1 - \frac{p}{a+bx}, 0\right)$$

So $x = 0$ is a self-fulfilling equilibrium if and only if $1 - \frac{p}{a} \leq 0$. Any other self-fulfilling belief must be a solution with $0 < x \leq 1$ to the quadratic equation:

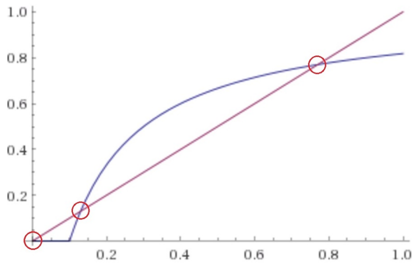


Figure 14.1: A graph (generated by Wolfram Alpha) of $y = F\left(\frac{p}{d(x)}\right)$ and $y = x$ for the example with $(a, b, p) = (0.1, 1, 0.2)$. The equilibria are characterized by the intersections. Also, notice that BRD will converge to a higher value when $F\left(\frac{p}{d(x)}\right) > x$ and lower when $F\left(\frac{p}{d(x)}\right) < x$; this allows us to characterize the stability of the equilibria.

$$1 - \frac{p}{a + bx} = x$$

which simplifies to

$$a + bx - p = ax + bx^2$$

or

$$x^2 + \frac{a-b}{b}x + \frac{p-a}{b} = 0$$

Then the solutions to this, which may be easily verified by the quadratic formula, are at

$$x = \frac{b-a}{2b} \pm \sqrt{\frac{a-p}{b} + \left(\frac{a-b}{2b}\right)^2}$$

So, even in this simple example, there can be between 0 and 3 self-fulfilling equilibria, depending on our choice of a, b, and p. For instance, setting $a = 0.1$, $b = 1$, $p = 0.2$ produces the example shown in Figure 14.1. $x_0 = 0$ is a self-fulfilling equilibrium, since $1 - \frac{p}{a} < 0$; the other two equilibria are given by the zeroes of the quadratic, $x_1 \approx 0.13$ and $x_2 \approx 0.77$.

Notice that these two outcomes are actually very different! Which of them will we arrive at assuming that people follow the best-response dynamics defined above? Clearly, if we start at an equilibrium, the BRD process will not move away from that. But if we move just a bit away from the equilibrium, it might turn out that BRD will converge to an entirely different equilibrium!

For instance, 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 call x_0 and x_2 *stable equilibria* and x_1 a *highly unstable equilibrium* for exactly this reason.

However, x_1 is still extremely important as a so-called "tipping point". If people believe that more than x_1 players will buy the good, we end up at the equilibrium x_2 where almost everyone buys it; if people believe that fewer than x_1 players will buy the good, then eventually the good will "die out" and *nobody* will buy it!

So, in order to be successful at selling their item, 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 actually 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, a seller should be willing to lower the price of their item, even below its production cost, until they achieve some appropriate penetration rate (above the corresponding "tipping point" to its actual production cost). Then they may increase the price back to the production cost.

14.2 Markets with Asymmetric Information

Self-fulfilling equilibria are also a useful tool to analyze markets with *asymmetric information*—that is, where the seller might have more information about the goods than the buyer. The following is a theory developed by George Akerlof and originally published in 1970.

Consider a market for used cars, where some of the cars are "lemons" which are worthless to the buyer and the seller alike, and others are good cars worth v_B to the buyer but $v_S < v_B$ to the seller. How much would a buyer be willing to pay for a random car, assuming that lemons and good cars are difficult to distinguish and that a fraction f of the cars are good?

If we consider a quasi-linear utility model, where a buyer's utility is their value for the item minus the price they pay for it, then a player wishing to ensure that their expected utility will be non-negative should be willing to pay at most $v_B x$, where x is their belief about the fraction of good cars *on the market* (note that not everyone necessarily puts their car up for sale).

So what should a seller do? If the seller has a lemon to sell, they should clearly put it up for sale, since they might be able to get money for it. And if

$v_B x > v_S$ they would be willing to put a good car up for sale; otherwise they would not.

Once again we are searching for a self-fulfilling equilibrium having the property that if everyone believes that a fraction x of the cars are good, then a fraction x of the cars in the market will be good.

For simplicity, consider a model where all sellers have the same value v_S and all buyers the same value v_B. Thus, either all sellers of good cars will put them up for sale, or *none* will; that is, $x = f$ or $x = 0$ are the only two candidates.

$x = 0$ is in fact always self-fulfilling—if everyone believes that the market consists entirely of lemons, nobody will be willing to pay anything for a car, and so there will never be any good cars put up for sale on the market, only lemons!

$x = f$, meanwhile, is self-fulfilling if and only if $v_B f > v_S$, or equivalently $f > \frac{v_S}{v_B}$.

We can again analyze BRD and notice that $\frac{v_S}{v_B}$ is a tipping point for the market. If *the sellers believe that the buyers believe* that a fraction of good cars smaller than this is on the market, they will not be willing to sell their good cars, and the market *crashes*.

Of course, it is unrealistic to assume that every car has the same value. But even if there are several different types of cars on the market, this effect will hold; if the market ever ends up falling under an appropriately-defined tipping point, then none of the best cars will enter the market, which will drive down the expected value of a car and in turn subsequently cause lower and lower valued cars to leave the market, until only lemons remain!

Finally, let us mention that this sort of market crash occurs not only in used car sales but also in labor markets where an employer might not be able to effectively evaluate the skills of a worker. Good workers are analogous to the good cars, and have some value they can generate on their own (e.g. being self-employed) and some value to the employer; bad workers are the "lemons" of the market, and the exact same analysis applies assuming the employer cannot distinguish the two types.

Spence's signalling model. *(Or: why university degrees are useful, even if you don't actually learn anything!)* One way out of this problem is to let players signal the value of their good. In the used car market, this tends to come in such forms as vehicle history reports and inspections. For the labor market, education can be a very useful 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 don't get degrees because the added salary isn't worth the extra cost of the degree to them, while it is worth it for good workers! So getting a degree may be worthwhile to the good workers (again, even if they don't gain any relevant skills or knowledge!), as their willingness to get the degree signals their higher value as an employee.

Andrew Spence, George Akerlof, and Joseph Stiglitz shared the 2001 Nobel Prize in Economic Sciences for their influential work on information and information dynamics in markets.