The Price of Anarchy
Olsthoorn, F.

The Price of Anarchy
Measuring the inefficiency of selfish networking

Master's thesis
Floris Olsthoorn
Thesis advisor: dr. F.M. Spieksma

Mathematisch Instituut, Universiteit Leiden

April 3, 2012

Contents

Introduction

The internet is operated by a huge number of independent institutions called transit ISPs (Internet Service Providers). They operate in their own interest, which often leads to great inefficiencies and instabilities in the global network. For example, ISPs tend to pass packets on to a neighboring ISP as quickly as possible, like a hot potato. This policy, called *early-exit routing*, is beneficial to the ISP, since it minimizes the load on its own network. However, it can greatly increase the total length of the path that a packet has to traverse to reach its goal [17]. So the selfish actions of the ISPs are detrimental to the welfare of the network as a whole.

In general, selfish users cause some measure of deviation from the theoretically optimal solution to networking problems. This thesis asks the question: how bad can this deviation get? Or more colourfully: what is the price of anarchy? In particular, we review some of the answers that have been proposed by mathematicians and computer scientist over the last decade or so. The answers provide a mixed message about selfish networking. In some cases selfish networking is guaranteed to deviate at most a small factor from the optimal solution. In some other cases, unfortunately, the deviation from the optimum caused by selfish users could be unbounded.

Game theory Before we rush to these answers, however, we must first make mathematical sense of the vaguely put question above. This thesis considers a game theoretic approach of research. A game is a purely mathematical object that models a situation where a group of players is confronted with strategic choices. If all players have chosen a strategy, then each player receives a pay-off dependent on the choices of himself and his competitors (and/or allies). The pay-off represents some value the players wish to maximize, such as profit, or minimize, such as latency experienced due to congestion.

A game doesn't tell us what strategies the players will choose. But this is exactly what we need to know, since we want to compare the solution found by selfish players to the optimal solution. Fortunately, in game theory there is a widely used predictor for what course of action the players will take. It is called the *Nash equilibrium*, which was first suggested by the mathematician John Forbes Nash in 1950 [9].

Suppose all players have chosen their strategy. If some player could profit by *unilaterally* changing his strategy and thereby getting a better pay-off, then obviously the player has the incentive to do just that. We are, therefore, not in a stable situation. If, on the other hand, no player can profit by unilaterally changing his strategy, the players are said to be in a Nash equilibrium[1].

The Price of Anarchy and Stability Once the players have settled on a choice of strategies, we would like to quantify the impact on the system as a whole. For this, we introduce a *social utility function*, which returns for each choice of strategies some number. This number could respresent, for example, the total profit gained by the players or the average latency each player experiences. Once we have defined the social utility function, we can compare the social utility of a Nash equilibrium with the value of optimal solutions. In this context an optimal solution is a choice of strategies that yields the best value for the social utility function.

The most popular metric for the impact of selfishness is called the *Price of Anarchy*. It is the proportion between the *worst possible* social utility from a Nash equilibrium and the optimal social utility, not necessarily from a Nash equilibrium. Notice that for this definition to make sense, a game should allow at least one Nash equilibrium. Not all games do, so for each game we need to check what, if any, the Nash equilibria are.

Sometimes we're also interested in the best-case scenario. We define the *Price of Stability* as the proportion between the *best possible* social utility of a Nash equilibrium and the optimal social utility. In other words, the Price of Stability measures how far we are from truly optimal when we reach the best possible solution that everyone can agree on.

Results In this thesis we find low constant bounds on the Price of Anarchy for several games. For example, any instance of the routing game discussed in Chapter 1 has a Price of Anarchy of at most 4/3, provided the latency functions of the edges in the network of the instance are affine. We prove a more general bound using elementary methods from continuous optimization. Specifically, readers familiar with Karush-Kuhn-Tucker theory should find the reasoning easy to follow.

Chapter 2 is dedicated to the analysis of network formation games. These are games where players build some type of network together, but they only want to maximize their own gain (or minimize their own cost). For example, in the Local Connection Game, players want to be closely connected to all other players, but want to build as few connections as possible, since each connection costs resources to build. The Price of Stability of any Local Connection Game is at most 4/3.

We prove that all games in Chapter 2, except the Local Connection Game, are examples of a special class of games called *potential games*. These games are studied in Chapter 3. Potential games allow a *potential function*, which is a single function that tracks the changes in utility as players change their strategies. The mere existence of such a function guarantees some powerful results. For example, potential games always have a (deterministic) Nash equilibrium. Also, *best-response dynamics*, where each turn one player changes to a strategy that maximizes his utility given the current strategies of the other players, will converge to a Nash equilibrium.

[1] Not all games have a Nash equilibrium. If players are allowed to have a 'mixed' strategy, i.e. choose each strategy with a certain probability, then a Nash equilibrium is guaranteed to exist, provided the amount of players and strategies are finite [9]. However, we will not consider mixed strategies in this thesis.

Algorithmic Game Theory and further research The study of the inefficiency of Nash equilibria is a topic in the broader field of *Algorithmic Game Theory*. This relatively young field aims to find constructive answers to questions that arise when one studies the internet. Some examples of topics studied in Algorithmic Game theory are online auctions, peer-to-peer systems and network design. For an overview of the field see [15]. The structure of this thesis is based largely on Chapters 17, 18 and 19 from [10], the first book on Algorithmic Game Theory. The book is freely available online.

This thesis only covers the fundamentals of the Price of Anarchy. We don't discuss such related topics as applications to network design, other equilibrium concepts and the computional aspects of finding Nash equilibria. The book [10] covers some of these topics. For a variation on the Price of Anarchy, see for instance [1]. This article defines the *strong equilibrium* in the context of job scheduling and network formation. The strong equilibrium is introduced to account for situations where players may form coalitions. It is a Nash equilibrium that is also resilient to deviations by coalitions. Even though the lack of coordination is resolved in a strong equilibrium, the (Strong) Price of Anarchy may still be larger than 1.

Chapter 1

Selfish routing

1.1 Introduction

This chapter focuses on Tim Roughgarden's work on *routing games* [14, 16], the 'paradigmatic' study in the area of Price of Anarchy [11]. The games are based on older models from transportation theory. A routing game consists of a network where players want to route traffic from a source to a destination. Each player chooses a path through the network connecting his source and destination. On each edge in his path, a player experiences latency dependent on the total amount of players who route their traffic along the same edge. The precise amount is determined by the *cost function* of the edge. The socially optimal solution to the routing problem is attained when the *total* latency experienced is minimal. Even in very simple networks the social optimum is not a 'selfish solution', i.e. a Nash equilibrium, as we will see in Example 1.3.1.

Most of this section focuses on the situation that the traffic is formed by a very large set of players, each of whom controls a negligible fraction of the traffic. We call this *nonatomic routing*. In this situation, the problem of determining the Price of Anarchy in any routing game has essentially been solved by Roughgarden. The Price of Anarchy of any given routing game depends *only* on the type of cost functions used. So other factors, such as the topology of the network or the distribution of sources and destinations, are irrelevant.

For any set of cost functions, there is a strict upper bound for the Price of Anarchy (Definition 1.6.1). For certain sets of functions this upper bound can be calculated explicitly. For example, if the cost functions are affine, then the price of anarchy is at most 4/3. This means that the total latency in any Nash equilibrium is at most 33% worse than in the optimal solution; a positive result indeed. On the other hand, if the cost functions are polynomial, then the Price of Anarchy becomes potentially unbounded.

In the closing section of this chapter, we will consider routing games with only a finite amount of players, each controlling a non-negligible amount of traffic, so called *atomic routing*. The analysis in this case is less 'clean' than in the nonatomic case. For instance, not all atomic routing games have a Nash equilibrium, in contrast with nonatomic routing. Still, we can deduce a positive result. Just like in nonatomic routing, if the cost functions are affine, then the Price of Anarchy is bounded by a small constant (~ 2.618, see Theorem 1.7.3).

1.2 The model

We will use a generalized version of the *Wardrop model* of transportation networks from [19]. In the model a flow is routed across a directed graph. This flow represents the traffic of a continuum of players, where each player controls an infinitessimal amount of traffic. This model is called **nonatomic routing**[1].

A **network** N is a directed finite graph $G = (V, E)$ together with $k_N \in \mathbb{Z}_{\geq 1}$ **commodities** $\{s_i, t_i\}$ where $s_i \in V$ is called the **source node** of the commodity and $t_i \in V$ is called the**sink node** of the commodity. Different commodities can share the same source or destination.

A **(nonatomic) instance** is a triple (N, r, c), where N is a network, r is a k-dimensional vector of **traffic rates** $r_i \in \mathbb{R}_{>0}$ for each commodity $\{s_i, t_i\}$ in N, and c is an $\#E$-dimensional vector of **cost functions** $c_e : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ for each edge e in N. A cost function is sometimes also called a **latency function** and measures the amount of latency per unit of traffic.

Let (N, r, c) be an instance. The set of $\{s_i, t_i\}$-paths, where $\{s_i, t_i\}$ is a commodity, is denoted by $\mathcal{P}_i$. The set $\mathcal{P}$ of commodity paths in N is defined by $\mathcal{P} = \cup_i \mathcal{P}_i$. A **flow** f on N is a vector in $\mathbb{R}_{\geq 0}^{\#\mathcal{P}}$, where f_P denotes the flow over path $P \in \mathcal{P}$. A flow f on N is **feasible** if $\sum_{P \in \mathcal{P}_i} f_P = r_i$ for each each commodity $\{s_i, t_i\}$. For each e the **flow on** e is given by

$$f_e = \sum_{P \in \mathcal{P} : e \in P} f_P.$$

We interpret the cost functions of an instance as measuring the cost or latency of an edge experienced by a unit of traffic. Thus, the latency experienced by the traffic f_e on edge e is $c_e(f_e)f_e$. The **cost of a flow** f on N is given by

$$C(f) = \sum_{e \in E} c_e(f_e) f_e.$$

A flow f_{opt} on N is considered **optimal** for the instance if f_{opt} is feasible and $C(f_{\text{opt}})$ is minimal, i.e.

$$C(f_{\text{opt}}) = \min\{C(f') : f' \text{ on } N \text{ is feasible}\}. \tag{1.2.1}$$

1.2.1 Remark. An objective function such as the total cost function defined above, where we sum the players' costs, is called a **utilitarian objective function**. Another type of objective function is the **egalitarian objective function**, which is often used in scheduling problems. This function is determined by the maximum of the players' costs.

We define the **cost of a path** $P \in \mathcal{P}$ by

$$c_P(f) = \sum_{e \in P} c_e(f_e).$$

Using this definition, we can rewrite (1.2.1) to

$$C(f) = \sum_{P \in \mathcal{P}} c_P(f) f_P.$$

[1] Atomic routing is discussed in Section 1.7

To model the behavior of players in a network, we assume the flow is in a sort of Nash equilibrium. Informally, a flow is considered to be at an equilibrium if no player can decrease his cost by *unilaterally* deciding to switch to another path (i.e. changing strategy while no other player changes strategy). This idea is captured in the following definition.

Let (N, r, c) be an instance and f a feasible flow on N. The flow f is a **Nash flow** if for every commodity $\{s_i, t_i\}$ and every pair of paths $P_1, P_2 \in \mathcal{P}_i$ where $f_{P_1} > 0$ the following condition holds:

$$c_{P_1}(f) \leq c_{P_2}(f).$$

We will see in Section 1.5 that all Nash flows have equal cost. Also, if $C(f_{\text{opt}}) = 0$, then f_{opt} is a Nash flow, because $c_P(f_{\text{opt}}) = 0$ whenever $(f_{\text{opt}})_P > 0$. These two facts justify the following definition of the Price of Anarchy. Let f_{opt} be an optimal flow and f_{Nash} a Nash flow for the instance (N, r, c). The **Price of Anarchy** of (N, r, c), denoted by $\rho(N, r, c)$ is defined by

$$\rho(N, r, c) = \frac{C(f_{\text{Nash}})}{C(f_{\text{opt}})},$$

where it is understood that $0/0 = 1$. For a set $\mathcal{I}$ of instances, the Price of Anarchy of $\mathcal{I}$, denoted by $\rho(\mathcal{I})$, is defined by

$$\rho(\mathcal{I}) = \sup_{(N,r,c)\in\mathcal{I}} \rho(N, r, c)$$

1.2.2 Remark. Note that a nonatomic instance is not modelled as a game, with a set of players, strategies and pay-offs. It is possible to model routing games in game theoretic terms equivalent to our model, but it is more convenient to formulate the important results without the added complexity of game theory.

1.3 Examples

We focus our attention on a specific example called the *Pigou network*. This seemingly innocuous example, with only two nodes and two edges connecting them, tells us practically everything we need to know about nonatomic routing games. We will prove in Section 1.6 that the Price of Anarchy of any nonatomic instance is determined by the worst-possible Pigou-like example which can be constructed with the cost functions of the instance.

1.3.1 Example (Pigou [13])**.** Consider the network N whith just two vertices s and t, two edges labeled 1 and 2 from s to t and one commodity $\{s, t\}$. Let $\text{Pig}(r, c)$ denote the instance (N, r, c). In this first example we take $r = 1$ (slight abuse of notation), cost functions as shown in Figure 1.1 and denote this instance with Pig.

- **Optimal flow:** Let f be a feasible flow for Pig. We must have $f_1 + f_2 = 1$, from which we derive $C_{\text{Pig}}(f) = f_1^2 - f_1 + 1$. Minimizing this function on $[0, 1]$ gives $C_{\text{Pig}}(f_{\text{opt}}) = 3/4$ with f_{opt} routing through each edge half of the traffic.

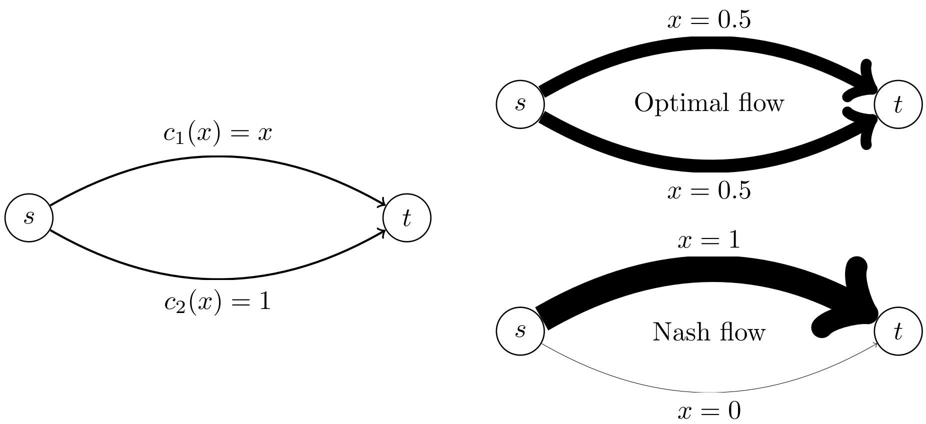

Figure 1.1: Pigou's example; a network with Price of Anarchy equal to 4/3. The image on the left shows the setup of the network, with the cost functions for the two edges. The variable $x \in [0, 1]$ represents the amount of traffic routed through the edge. The top right image shows the optimal flow, with a total cost of $3/4 (= 0.5 \cdot 0.5 + 0.5 \cdot 1)$. This is not a Nash flow, since the traffic at edge 2 experiences higher latency than the traffic at edge 1. The bottom right image shows the only Nash flow for Pigou's network, with a total cost of 1.

- **Nash flow:** Consider the flow f_{Nash} routing all traffic through 1 (i.e. $(f_{\text{Nash}})_1 = 1$ and $(f_{\text{Nash}})_2 = 0$). Given the flow f_{Nash}, the two paths in Pig have equal cost, namely 1. Therefore f_{Nash} is a Nash flow with cost 1.
- **PoA:** The Price of Anarchy of Pig is therefore $\rho(\text{Pig}) = 4/3$.

This nice result on the Price of Anarchy in Pigou's network holds for any *affine cost function*, i.e. a cost function of the form $c(x) = ax + b$, where $a,b \in \mathbb{R}_{\geq 0}$. Before we prove this, we first examine a general version of the Pigou network.

1.3.2 Example (Pigou (general))**.** In this more general version of the Pigou example, we take the same network N as in Example 1.3.1, but we consider an arbitrary traffic rate $r \in \mathbb{R}_{>0}$, an arbitrary cost function $c(x)$ for edge 1 and set $c_2(x) = c(r)$.

- **Optimal flow:** A feasible flow f for $\text{Pig}(r, c)$ routes a certain amount of traffic x through edge 1 and the remaining amount of traffic $r - x$ through edge 2. The cost of such a flow is given by $xc(x) + (r - x)c(r)$. The cost of an optimal flow, then, is given by

$$C_{\text{Pig}(r,c)}(f_{\text{opt}}) = \inf_{x \in [0,r]} xc(x) + (r - x)c(r).$$

- **Nash flow:** Consider the flow f_{Nash} routing all traffic through 1 (i.e. $(f_{\text{Nash}})_1 = 1$ and $(f_{\text{Nash}})_2 = 0$). Given the flow f_{Nash}, the two paths in Pig have equal cost, namely $c(r)$. Therefore f_{Nash} is a Nash flow with cost $r \cdot c(r)$.

- **PoA:** The Price of Anarchy of $\mathrm{Pig}(r,c)$ is therefore

$$\rho(\mathrm{Pig}(r,c)) = \sup_{x\in[0,r]} \frac{r\cdot c(r)}{xc(x)+(r-x)c(r)}. \tag{1.3.1}$$

For certain functions the expression on the right-hand side of (1.3.1) can be rewritten to a somewhat nicer form. For example, a simple calculation for the affine cost function $c(x) = ax + b$, where $a,b \in \mathbb{R}_{\geq 0}$, shows that the Price of Anarchy is at worst the same as in Example 1.3.1:

$$\begin{aligned}
\rho(\mathrm{Pig}(r,c)) &= \sup_{x\in[0,r]} \frac{r(ar+b)}{x(ax+b)+(r-x)(ar+b)} \\
&= \sup_{x\in[0,r]} \frac{r(ar+b)}{ax^2-arx+r(ar+b)} \\
&= \frac{r(ar+b)}{a(r/2)^2-ar(r/2)+r(ar+b)} \\
&= \frac{r(ar+b)}{(3/4)r(ar+b)+(r/4)b} \\
&\leq \frac{4}{3}.
\end{aligned}$$

The above result also holds for any concave cost function, since any concave function c can be bounded from below by an affine cost function that agrees with c on $x = r$, namely $c'(x) = (c(r)/r)x$. Note that from equation (1.3.1) it follows that if c and c' are cost functions with $c(r) = c'(r)$ and $c(x) \geq c'(x)$ for each $x \in [0,r]$, then $\rho(\mathrm{Pig}(r,c)) \leq \rho(\mathrm{Pig}(r,c'))$. Consequently, $\rho(\mathrm{Pig}(r,c)) \leq \rho(\mathrm{Pig}(r,c')) \leq 4/3$.

The nice bound on the Price of Anarchy found in Example 1.3.1 evaporates as soon as we introduce some form of nonlinearity. For example, if the cost function c satisfies $c(x) = x^p$, with $p \in \mathbb{Z}_{>0}$, then $\rho(\mathrm{Pig}(r,c))$ grows to infinity as p goes to infinity. Indeed, in this case the denominator of the right-hand side of equation (1.3.1) is minimized at $x = r(p+1)^{-1/p}$. Therefore the Price of Anarchy satisfies

$$\begin{aligned}
\rho(\mathrm{Pig}(r,c)) &= \sup_{x\in[0,r]} \frac{r^{p+1}}{x^{p+1}+(r-x)r^p} \\
&= \frac{r^{p+1}}{r^{p+1}\left((p+1)^{-(p+1)/p}+1-(p+1)^{-1/p}\right)} \\
&= \frac{1}{1-p(p+1)^{-(p+1)/p}}.
\end{aligned}$$

Since

$$\lim_{p\to\infty} p(p+1)^{-(p+1)/p} = \lim_{p\to\infty} \frac{p}{p+1} \cdot \lim_{p\to\infty} \frac{1}{(p+1)^{1/p}} = 1\cdot 1 = 1,$$

the Price of Anarchy goes to infinity as p goes to infinity.

1.4 Equivalence of optimal and Nash flows

Although the definitions of optimal and Nash flows are quite different—one minimizing the total cost incurred by the players, the other requiring that all paths with positive flow have equal cost—, a striking correspondence exists between the two types of flow. In particular, the optimal flow in an instance is a Nash flow in a closely related instance. From this correspondence we can derive existence and (essential) uniqueness results for Nash flows in nonatomic instances (Section 1.5).

We begin this section by finding characterizations of an optimal flow. One of these characterizations looks suspiciously similar to the definition of a Nash flow. This will inspire the equivalence between optimal and Nash flows which we will derive at the end of the section.

Given a nonatomic instance (N, r, c) the problem of finding a feasible, optimal flow f is the same as the convex program

$$\begin{aligned}
&\min \sum_{e \in E} h_e(f_e) && \\
&\text{subject to} \sum_{P \in \mathcal{P}_i} f_P = r_i && \text{for all } 1 \leq i \leq k \\
&f_e = \sum_{P \in \mathcal{P}: e \in P} f_P && \text{for all } e \in E \\
&f_P \geq 0 && \text{for all } P \in \mathcal{P} \\
&f \in \mathbb{R}^{\#\mathcal{P}} && ,
\end{aligned} \tag{1.4.1}$$

where $\sum_{e \in E} h_e$, the **objective function** of (1.4.1), is given by $h_e(f_e) = c_e(f_e) f_e$. The set of all flows f which satisfy the constraints in (1.4.1) is called the **feasible region** of (1.4.1). Note that since all constraints are linear, the feasible region of (1.4.1) is convex. Let h'_e denote the derivate $\frac{d}{dx} h_e(x)$ of h_e and $h'_P(f)$ denote the sum $\sum_{e \in P} h'_e(f_e)$.

1.4.1 Theorem (Characterization of optimal flows)**.** *Consider the nonlinear program* (1.4.1)*. Let f be a solution to this program. Suppose that every h_e is continuously differentiable and convex. The following are equivalent:*

(a) The flow f is optimal.

(b) For every $1 \leq i \leq k$ and every pair of paths $P_1, P_2 \in \mathcal{P}_i$ where $f_{P_1} > 0$ the following condition holds:

$$h'_{P_1}(f) \leq h'_{P_2}(f).$$

Proof. Since the objective function is continuously differentiable and convex, the flow f is optimal if and only if it satisfies the so-called Karush-Kuhn-Tucker (KKT) conditions [12]*Corollary 3.20. Denote the objective function and constraints of (1.4.1) as follows:

$$\begin{aligned} C(f) &= \sum_{e \in E} h_e(f_e) \\ h_i(f) &= \left(\sum_{P \in \mathcal{P}_i} f_P \right) - r_i \quad 1 \le i \le k \\ g_P(f) &= -f_P \end{aligned}$$

Then f is optimal if and only if there exist $\mu_P \in \mathbb{R}$, for each $P \in \mathcal{P}$ and $\lambda_i \in \mathbb{R}$ for each $1 \le i \le k$ such that the KKT-conditions are satisfied:

$$\begin{aligned} \nabla C(f) + \sum_{P \in \mathcal{P}} \mu_P \nabla g_P(f) + \sum_{i=1}^{k} \lambda_i \nabla h_i(f) = 0 \\ g_P(f) \le 0, \quad & \text{for each } P \in \mathcal{P} \\ h_i(f) = 0, \quad & \text{for each } 1 \le i \le k \\ \mu_P \ge 0, \quad & \text{for each } P \in \mathcal{P} \\ \mu_P g_P(f) = 0, \quad & \text{for each } P \in \mathcal{P} \end{aligned}$$

Suppose f is optimal. Let μ_P, $P \in \mathcal{P}$ and λ_i, $1 \le i \le k$ be such that f satisfies the KKT-conditions. For each $P \in \mathcal{P}$ we have

$$\begin{aligned} (\nabla C(f))_P &= \sum_{e \in P} h'_e(f_e) = h'_P(f) \\ (\nabla g_{\tilde{P}}(f))_P &= \begin{cases} -1 & \text{if } \tilde{P} = P \\ 0 & \text{if } \tilde{P} \ne P \end{cases} \\ (\nabla h_i(f))_P &= \begin{cases} 1 & \text{if } P \in \mathcal{P}_i \\ 0 & \text{if } P \notin \mathcal{P}_i \end{cases}. \end{aligned}$$

So for $P \in \mathcal{P}_i$ we have

$$\begin{aligned} h'_P(f) - \mu_P + \lambda_i = 0 \\ \mu_P \ge 0, -\mu_P f_P = 0 \end{aligned}$$

Consider P_1,$P_2 \in \mathcal{P}_i$ where $f_{P_1} > 0$. Then $\mu_{P_1} = 0$, so

$$\begin{aligned} h'_{P_1}(f) &= -\lambda_i \\ h'_{P_2}(f) &= -\lambda_i + \mu_{P_2}. \end{aligned}$$

Since $\mu_{P_2} \ge 0$, it follows that $h'_{P_1}(f) \le h'_{P_2}(f)$, which we wanted to prove.

Suppose that condition (b) from the statement of the theorem holds. Since f is feasible, for each $1 \le i \le k$, there is a path $P \in \mathcal{P}_i$ such that $f_P > 0$. Also, for each $1 \le i \le k$, every pair of paths P_1, $P_2 \in \mathcal{P}_i$ with f_{P_1},$f_{P_2} > 0$ must satisfy $h'_{P_1}(f) = h'_{P_2}(f)$. For each $1 \le i \le k$ and each $P \in \mathcal{P}_i$, set:

$$\begin{aligned} \lambda_i &= -h_{P_i}(f) \\ \mu_P &= h'_P(f) - h'_{P_i}(f), \end{aligned}$$

where $P_i \in \mathcal{P}_i$ is such that $f_{P_i} > 0$. With these constants the KKT-conditions are satisfied, as can be easily verified. It follows that f is optimal. □

Theorem 1.4.1 says that finding an optimal solution for an instance (N, r, c) is the same as finding a Nash equilibrium in the same instance, but with different cost functions. More precisely, given a cost function $c_e(f_e)$, we call $c_e^*(f_e) := (f_e \cdot c_e(f_e))'$ the **marginal cost function** for the edge e. Then Theorem 1.4.1 immediatly implies the following corollary:

1.4.2 Corollary. *Let (N, r, c) be an instance such that each function $f_e \cdot c_e(f_e)$ is continously differentiable and convex. Then f is an optimal flow for (N, r, c) if and only if f is a Nash flow for (N, r, c^*).*

Notice that Corollary 1.4.2 works the other way too. Indeed, suppose we want to find a Nash flow f for (N, r, c). For each edge e define $h_e(f_e) = \int_0^{f_e} c_e(x)dx$. Since each c_e is continuous and nondecreasing, each h_e is continuously differentiable and convex. Moreover, each h_e satisfies $h_e'(f_e) = c_e(f_e)$. So we can apply Theorem 1.4.1 by considering the nonlinear program with the following objective function:

$$\Phi(f) = \sum_{e \in E} \int_0^{f_e} c_e(x)dx. \tag{1.4.2}$$

We call (1.4.2) the **potential function** of the nonatomic instance (N, r, c). This yields the following corollary:

1.4.3 Corollary. *Let (N, r, c) be a nonatomic instance. A feasible flow f for (N, r, c) is a Nash flow precisely when it minimizes Φ on the set of feasible flows for (N, r, c).*

1.4.4 Remark. Changing the cost functions of an instance (N, r, c) does not change the set of feasible flows. This allows us to apply Theorem 1.4.1 as in Corollary 1.4.3.

We close this section by proving another characterization of Nash flows, the *variational inequality characterization*, using Corollary 1.4.3. We use this technical result to find a strict upper bound on the Price of Anarchy of nonatomic instances (Theorem 1.6.3).

1.4.5 Corollary. *Let (N, r, c) be a nonatomic instance. A feasible flow f for (N, r, c) is a Nash flow precisely when for every feasible flow f^* for (N, r, c), the following inequality holds:*

$$\sum_{e \in E} c_e(f_e)f_e \leq \sum_{e \in E} c_e(f_e)f_e^*. \tag{1.4.3}$$

Proof. We apply Corollary 1.4.3 for this proof. Let f and f^* be feasible flows for (N, r, c). For each $e \in E$ the following inequality holds:

$$c_e(f_e)(f_e^* - f_e) \leq \int_0^{f_e^*} c_e(x)dx - \int_0^{f_e} c_e(x)dx. \tag{1.4.4}$$

This is due to the cost functions being nondecreasing. So if (1.4.3) holds for each feasible flow f^*, then f minimizes the potential function Φ of (N, r, c) and is therefore optimal.

For the reverse implication, let f^* be a feasible flow for which (1.4.3) doesn't hold, i.e.

$$\sum_{e\in E} c_e(f_e)(f_e^* - f_e) < 0.$$

For each $\lambda \in [0,1]$ consider the flow f^λ given by $f_e^\lambda = \lambda f_e^* + (1-\lambda) f_e$ for each $e \in E$. All these flows are feasible[2] and don't satisfy (1.4.3):

$$\sum_{e\in E} c_e(f_e)(f_e^\lambda - f_e) = \lambda \sum_{e\in E} c_e(f_e)(f_e^* - f_e) < 0.$$

As λ approaches 0, the difference of integrals on the right-hand side of (1.4.4) approaches $c_e(f_e)(f_e^\lambda - f_e)$ much faster than $c_e(f_e)(f_e^\lambda - f_e)$ approaches 0. Indeed, using the first-order Taylor approximation we get

$$\lim_{\lambda\to 0} \frac{\int_0^{f_e^\lambda} c_e(x)dx - \int_0^{f_e} c_e(x)dx}{c_e(f_e)(f_e^\lambda - f_e)} = \frac{1}{c_e(f_e)} \frac{d}{dy} \int_0^y c_e(x)dx \bigg|_{y=f_e} = 1.$$

So if we take λ close enough to 0, we get

$$\sum_{e\in E} \int_0^{f_e^\lambda} c_e(x)dx - \int_0^{f_e} c_e(x)dx < 0.$$

By Corollary 1.4.3, f is not optimal. □

1.5 Existence and uniqueness of flows

With the tools of Section 1.4 in our arsenal, it is a relatively straightforward affair to prove that in each nonatomic instance, there exists an equilibrium flow and it is (essentially) unique.

1.5.1 Theorem. *Let (N, r, c) be a nonatomic instance. There exists a Nash flow. If f and f^* are Nash flows for (N, r, c), then $c_e(f_e) = c_e(f_e^*)$ for each edge $e \in E$.*

Proof. Recall that the set of feasible flows of (N, r, c) contains all flows $f \in \mathbb{R}^{\#\mathcal{P}}$ for which $f_P \geq 0$ for each $P \in \mathcal{P}$ and $\sum_{P\in\mathcal{P}_i} f_P = r_i$ for each commodity i. This set is compact in $\mathbb{R}^{\#\mathcal{P}}$. Consequently, the (continuous) potential function (1.4.2) of (N, r, c) attains a minimum at the set of feasible flows. By Corollary 1.4.3, the feasible flow at which the potential function is minimized, is a Nash flow.

Suppose f and f^* are Nash flows for (N, r, c). They both minimize the potential function Φ of (N, r, c). Since each h_e is convex, for each convex combination

[2] For each commodity $\{s_i, t_i\}$ we have

$$\sum_{P\in\mathcal{P}_i} f_P^\lambda = \lambda \sum_{P\in\mathcal{P}_i} f_P^* + (1-\lambda) \sum_{P\in\mathcal{P}_i} f_P = \lambda r_i + (1-\lambda) r_i = r_i$$

$f^\lambda = \lambda f + (1-\lambda)f^*$, where $\lambda \in [0,1]$, we have[3]

$$\begin{aligned}\Phi(f) \le \Phi(f^\lambda) &= \sum_{e\in E} h_e(f_e^\lambda)\\ &\le \sum_{e\in E} \lambda h_e(f_e) + (1-\lambda)h_e(f_e^*)\\ &= \Phi(f).\end{aligned}$$

For each h_e this yields

$$h_e(f_e^\lambda) = \lambda h_e(f_e) + (1-\lambda)h_e(f_e^*),$$

i.e. each h_e is linear between f_e and f_e^*. Consequently, each cost function c_e is constant between f_e and f_e^*. □

1.6 The Price of Anarchy of Routing Games

We promised in Section 1.3 that the Pigou network, a simple network with just one commodity, two edges and two vertices, would tell us basically everything we needed to know about the Price of Anarchy in nonatomic instances. In this section, we deliver on this promise. The central result of this section is that, given (almost) any constraint on the set of allowable cost functions, the worst Price of Anarchy of a Pigou network satisfying the constraint on the cost functions is also the worst possible Price of Anarchy of any instance satisfying the constraint.

Given a non-empty set $\mathcal{C}$ of cost functions, let $\rho(\mathcal{C})$ denote the supremum over all $\rho(I)$, where I is a nonatomic instance with cost functions in $\mathcal{C}$. We will prove that if $\mathcal{C}$ contains all constant functions, $\rho(\mathcal{C})$ is equal to the *Pigou bound*:

1.6.1 Definition. Let $\mathcal{C}$ be a non-empty set of cost functions. The **Pigou bound** for $\mathcal{C}$, denoted by $\alpha(\mathcal{C})$, is

$$\begin{aligned}\alpha(\mathcal{C}) &= \sup_{c\in\mathcal{C}} \sup_{r\ge 0} \rho(\mathrm{Pig}(r,c))\\ &= \sup_{c\in\mathcal{C}} \sup_{x,r\ge 0} \frac{r\cdot c(r)}{xc(x)+(r-x)c(r)},\end{aligned}$$

where we let $0/0$ take the value 1.

1.6.2 Remark. Taking the supremum over $x \ge 0$ is the same as taking the supremum over $x \in [0,r]$, since all cost functions are increasing.

1.6.3 Theorem. *Let $\mathcal{C}$ be a set of cost functions containing all the constant functions. Then $\rho(\mathcal{C}) = \alpha(\mathcal{C})$.*

Proof. From the definition of $\alpha(\mathcal{C})$ it follows that for any $\eta < \alpha(\mathcal{C})$ there is some instance $\mathrm{Pig}(r,c)$ with $\rho(\mathrm{Pig}(r,c)) > \eta$. Consequently, $\rho(\mathcal{C}) \ge \alpha(\mathcal{C})$.

We prove the other inequality using the variational inequality characterization of Nash flows, Corollary 1.4.5. Let (N,r,c) be a nonatomic instance with cost function in $\mathcal{C}$ and let f^* and f be optimal and Nash flows, respectively, for

[3] For each $\lambda \in [0,1]$, f^λ is a feasible flow.

this instance. First notice that for each edge $e \in E$, the following inequality holds:

$$\alpha(\mathcal{C}) \geq \frac{f_e \cdot c_e(f_e)}{f_e^* c_e(f_e^*) + (f_e - f_e^*)c_e(f_e)}.$$

Applying this inequality and Corollary 1.4.5 to the Price of Anarchy of (N, r, c) yields:

$$\begin{aligned} C(f^*) &= \sum_{e \in E} c_e(f_e^*) f_e^* \\ &= \sum_{e \in E} c_e(f_e) f_e \cdot \frac{f_e^* c_e(f_e^*) + (f_e - f_e^*)c_e(f_e)}{f_e \cdot c_e(f_e)} + c_e(f_e)(f_e^* - f_e) \\ &\geq \frac{1}{\alpha(\mathcal{C})} \sum_{e \in E} c_e(f_e) f_e + \sum_{e \in E} c_e(f_e)(f_e^* - f_e) \\ &\geq \frac{C(f)}{\alpha(\mathcal{C})}. \end{aligned}$$

In conclusion $\rho(\mathcal{C}) \leq \alpha(\mathcal{C})$. □

1.7 Atomic Routing

1.7.1 Introduction

Suppose only a finite amount of players route traffic across an instance (N, r, c). Each player controls a finite, but non-negligible, amount of traffic. Then this 'atomic' instance deviates from the nonatomic one in a couple notable respects. Firstly, different Nash flows may have different cost. Secondly, an atomic instance does not necessarily have a Nash flow. Thirdly, the Price of Anarchy of an atomic instance may be worse than in the nonatomic version of the instance. However, if the cost functions are affine, then a Nash flow always exists and there's good news as in the nonatomic case. The Price of Anarchy of an atomic instance with affine cost functions is at most a constant: $(3 + \sqrt{5})/2$. We will prove these facts in this section, but first we define the model.

1.7.2 The model

An atomic instance (N, r, c) is a finite strategic game based on the instance (N, r, c) as defined in Section 1.2. The player set is $A = \{1, \ldots, k_N\}$, where player i is associated with commodity $\{s_i, t_i\}$ and has traffic rate r_i. If all r_i are equal, we call the instance **unweighted**. The **strategy set** of player i is $\mathcal{P}_i$. We denote $\mathcal{S} = \Pi_{i=1}^{k_N} \mathcal{P}_i$. A flow $f \in \mathbb{R}_{\geq 0}^{\#\mathcal{P}}$ is called a **feasible flow** if there is an $s \in \mathcal{S}$ such that, for each $P \in \mathcal{P}$,

$$f_P = \sum_{i \in A : s_i = P} r_i.$$

This means that if player i chooses path P, then it routes r_i amount of traffic through path P. Player i's **cost function**, given a strategy s with corresponding flow f, is

$$\text{Cost}_i(f) = r_i \cdot c_{s_i}(f) = r_i \cdot \sum_{e \in s_i} c_e(f_e),$$

The **cost of a flow** f is given by

$$\text{Cost}(f) = \sum_{i=1}^{k_N} \text{Cost}_i(f) = \sum_{e \in E} c_e(f_e) f_e.$$

Each player wished to minimize his cost. This means the Nash equilibrium in an atomic instance is defined as follows. A feasible flow f is called a **Nash flow** if, for each player i and each pair of paths P,$P' \in \mathcal{P}_i$ such that $f_P > 0$, the following inequality holds:

$$c_P(f) \leq c_{P'}(f'),$$

where f' is the feasible flow equal to f, except $f'_P = f_P - r_i$ and $f'_{P'} = f_P + r_i$. (The traffic rate r_i is on both sides of the inequality, so can be cancelled out.)

1.7.3 Results

1.7.1 Theorem (from [3]). *Nash flows in an atomic instance do not always have the same cost.*

Proof. An example of an atomic instance proving this statement is called the AAE example (named after its discoverers). It is shown in Figure 1.2. □

1.7.2 Theorem (from [5]). *There is an atomic instance for which no Nash flow exists.*

Proof. An example of such an instance is shown in Figure 1.3. For any strategy pair (P_i, P_j), where P_i is the path chosen by player 1 and P_2 the path chosen by player 2, one of the players can improve his outcome by choosing another path. Indeed, we have the following unique **best responses** (a strategy change by one player that minimizes his cost) for each strategy pair:

$$\begin{aligned}
&(P_i, P_j) \to (P_3, P_j), j \in \{1, 2\} \\
&(P_i, P_j) \to (P_1, P_j), j \in \{3, 4\} \\
&(P_3, P_j) \to (P_3, P_4) \\
&(P_1, P_j) \to (P_1, P_2).
\end{aligned}$$

All the best responses shown are strict improvements for the player who changes his strategy. This shows that no strategy pair forms a Nash flow. □

If an atomic instance is unweighted *or* all cost functions are affine, then the instance admits a potential function. We encountered such a function earlier in the nonatomic case (function (1.4.2)). A potential function 'tracks' the cost increase experienced by a player when he unilaterally changes his strategy. The existence of such a function guarantees the existence of a Nash equilibrium. In Chapter 3 we study games that admit potential functions in detail.

1.7.3 Theorem. *Let (N, r, c) be an atomic instance with affine cost functions. Then*

$$\rho(N, r, c) \leq \frac{(3 + \sqrt{5})}{2} (\simeq 2.618)$$

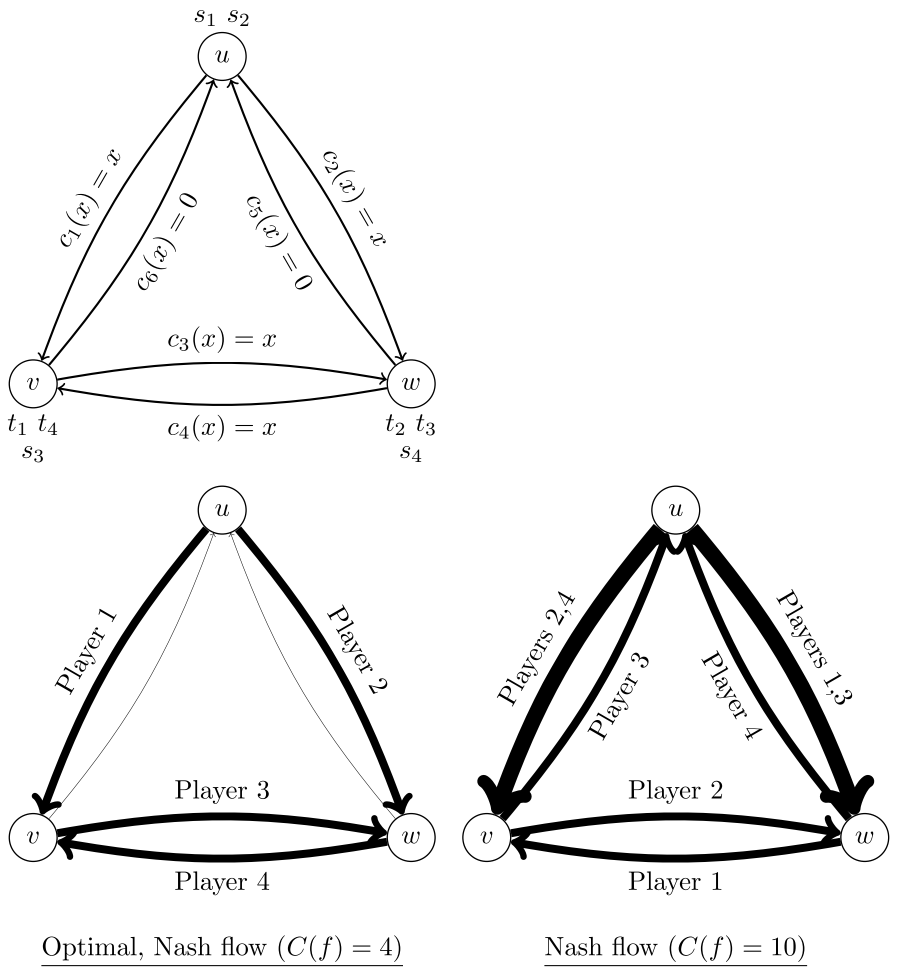

Figure 1.2: AAE example; an atomic instance with two Nash flows that have different costs. This situation cannot occur in nonatomic instances. Each player has the same traffic rate, $r = 1$, and the Price of Anarchy is $5/2$. If we set the traffic rate for players 1 and 2 to $\frac{1}{2}(1 + \sqrt{5})$ (the golden ratio) and the traffic rate for the other players to 1, then the Price of Anarchy is $\frac{1}{2}(3 + \sqrt{5})$, which is the highest possible Price of Anarchy for atomic instances with affine cost functions (Theorem 1.7.3).

Proof. Let f be a Nash flow and g an optimal flow for (N, r, c). We need to prove that

$$\frac{\text{Cost}(f)}{\text{Cost}(g)} \leq \frac{3 + \sqrt{5}}{2}.$$

For each edge $e \in E$, let $c_e(x) = a_e x + b_e$, $a_e, b_e \geq 0$ denote the cost function for edge e. Let player i choose path P_i in f and P'_i in g. From the definition of

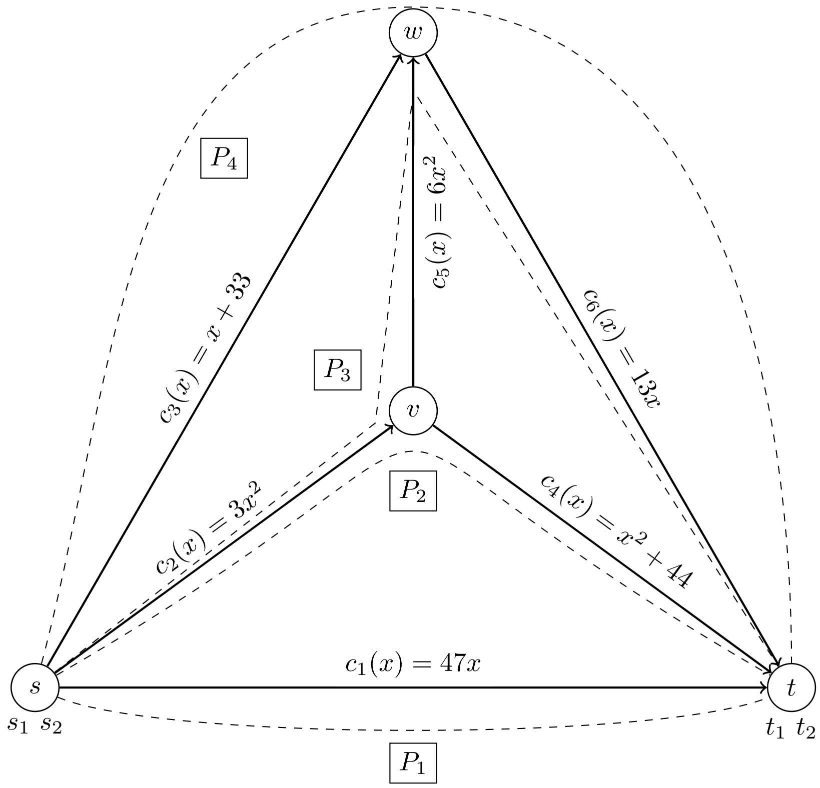

Figure 1.3: An atomic instance without a Nash flow. There are two players $\{1, 2\}$, each with commodity $\{s, t\}$ and traffic rates $r_1 = 1$ and $r_2 = 2$. If this example would have only affine cost functions or be unweighted (all traffic rates the same), then a Nash flow would exist.

Nash flow it immediatly follows that

$$\sum_{e \in P_i} a_e f_e + b_e \leq \sum_{e \in P_i'} a_e(f_e + r_i) + b_e.$$

Therefore, we have the following inequality:

$$\begin{aligned}
\text{Cost}(f) \leq \sum_{i=1}^{k_N} r_i \sum_{e \in P_i'} a_e(f_e + r_i) + b_e &\leq \sum_{i=1}^{k_N} r_i \sum_{e \in P_i'} a_e(f_e + g_e) + b_e \\
&= \sum_{e \in E(N)} g_e(a_e(f_e + g_e) + b_e) \\
&= \text{Cost}(g) + \sum_{e \in E(N)} a_e f_e g_e.
\end{aligned}$$

To the sum in the last line above we apply the Cauchy-Schwartz inequality:

$$\sum_{e \in E} a_e f_e g_e \leq \sqrt{\sum_{e \in E} a_e f_e^2} \cdot \sqrt{\sum_{e \in E} a_e g_e^2} \leq \sqrt{\text{Cost}(f)} \cdot \sqrt{\text{Cost}(g)}.$$

We combine the two inequalities we have derived so far to yield the following:

$$\left(\frac{\mathrm{Cost}(f)}{\mathrm{Cost}(g)} - 1\right)^2 \leq \frac{\mathrm{Cost}(f)}{\mathrm{Cost}(g)}.$$

The solution to this quadratic inequality is given by

$$\frac{\mathrm{Cost}(f)}{\mathrm{Cost}(g)} \leq \frac{3+\sqrt{5}}{2}.$$

□

Chapter 2

Network formation

2.1 Introduction

In Chapter 1, players are the users on an already existing network. In this chapter, players need to build the network themselves. One can think of real-life examples such as the physical creation of a network by ISPs or, more abstractly, the establishing of connections between users in peer-to-peer networks.

The games in this chapter all have some financial aspect. For example, in the Local Connection Game from Section 2.2, players have to pay for the connections that they decide to build. In the Facility Location Game from Section 2.4, players want to service customers in such a way that they earn the highest total price.

The games are simple, but the analysis becomes quite complex in some cases. Especially for the Local Connection Game we need to delve deep into graph theoretical technicalities to prove a bound on the Price of Anarchy. Nevertheless, the proofs require only little prior knowledge.

All games in this chapter, save for the Local Connection Game, are instances of *potential games*, i.e. games that admit a *potential function*. We already encountered a potential function when we characterized Nash flows for nonatomic routing games (Section 1.4). Potential games are described in detail in Chapter 3.

2.2 Local Connection Game

2.2.1 Introduction

In the Local Connection Game, introduced by Alex Fabrikant et al. in [4], a group of players seek to be connected to each other by building edges between them. The edges could, for example, represent friendships between people or landlines between servers. A player is connected to another player if there is a path of edges from him to the other player. The players want these paths to be as short as possible.

Of course, a player can easily minimize the lengths of the paths to other players by constructing all possible edges between him and each other player. However, the edges all have a fixed construction cost α. Perhaps it becomes profitable to depend on the edges built by other players, thereby saving in building costs. However, any decrease in construction costs could result in an increase in *usage costs*, i.e. an increase in distances to other players, where each increase of one edge represents a usage cost increase of 1.

In this section we explore the interplay between the conflicting interests of minimizing usage costs and constructing costs. We try to analyze how Nash equilibria compare to a situation where the *social cost* is minimized, i.e. where the sum total of the players' construction and usage costs is minimized, disregarding individual players' considerations.

In the best case scenario, when a Nash equilibrium results in a social cost as close as possible to the minimal social cost, the selfishly chosen strategies are indeed socially optimal for most values α. Only when α lies between 1 and 2 do the selfish users deviate from the socially optimal situation, but only by a factor of at most 4/3 (Theorem 2.2.3).

In the worst case scenario—the Price of Anarchy—the specific value of α has a lot of influence on what Nash equilibria look like. If α is very small, specifically if $\alpha < 1$, then the only Nash equilibrium is the complete graph. This also happens to be the optimal solution, so the Price of Anarchy in this case is 1. The proof of this result is only a few lines long. If α is very large, i.e. when $\alpha > 273n$, where n is the number of players, then all Nash equilibria are trees. This is fortunate, since trees are guaranteed to have a Price of Anarchy of less than 5. The proof of this result, however, is by far the longest, most technical proof in this thesis.

2.2.2 The model

An **instance** $\mathrm{LCG}(n, \alpha)$, where $n \in \mathbb{Z}_{>1}$ and $\alpha \in \mathbb{R}_{>0}$, of the Local Connection Game is a finite strategic game. The player set is $A = \{1, \ldots, n\}$. The strategy set for player i is $S_i = \mathcal{P}(A \setminus \{i\})$. A strategy vector $s \in \Pi_{i=1}^{n} S_i$ generates an undirected graph called a **network** $N(s) = (V(s), E(s))$ as follows. The set of nodes is $V(s) = A$. An edge $\{i, j\}$ is in $E(s)$ if and only if $i \in s_j$ or $j \in s_i$. We say that player i builds edge $\{i, j\}$ if $j \in s_i$.

Let s be a strategy vector for $\mathrm{LCG}(n, \alpha)$. Player i incurs a **construction cost** $c_i(s)$ for the edges that he builds. The construction cost is given by

$$c_i(s) = \alpha |s_i|.$$

Player i also experiences a **usage cost** $u_i(s)$ dependent on his proximity to the other players in the network $N(s)$. It is given by

$$u_i(s) = \sum_{j=1}^{n} \mathrm{dist}(i, j). \tag{2.2.1}$$

where $\mathrm{dist}(i, j)$ is the length of the shortest path (in terms of number of edges) from i to j. If there is no path from i to j, $\mathrm{dist}(i, j)$ is set to infinity. The **cost function** $\mathrm{Cost}_i(s)$ of player i is simply the sum of his construction and usage costs:

$$\mathrm{Cost}_i(s) = c_i(s) + u_i(s).$$

The **social cost function** $\mathrm{Cost}(s)$ is the sum of the players' costs:

$$\mathrm{Cost}(s) = \sum_{i=1}^{n} \mathrm{Cost}_i(s).$$

Nash equilibirum, optimal strategy and the Price of Anarchy and Stability are defined as in Appendix B. Note that both optimal strategies and Nash equilibria always form connected graphs. This is because disconnected graphs have infinite cost, so there is at least one player that can improve his own cost as well as the social cost by building all the edges connecting him to the other players. It follows that networks of $\mathrm{LCG}(n, \alpha)$ have at least $n - 1$ edges if they are generated by an optimal strategy or a Nash equilibrium.

A strategy s where two players build the same edge (by including each other in their strategy sets) is neither a Nash equilibrium nor optimal. We will therefore henceforth assume at most one player builds the same edge. This means that the social cost of s can be expressed as

$$\mathrm{Cost}(s) = \alpha|E(s)| + \sum_{i=1}^{n}\sum_{j=1}^{n} \mathrm{dist}(i, j).$$

So the social cost of s is only dependent on the structure of the generated network $N(s)$, not on which player builds what edge. This justifies the following definition. The **social cost** of an undirected graph G, denoted by $\mathrm{Cost}(G)$, is defined as

$$\mathrm{Cost}(G) = \mathrm{Cost}(s),$$

where s is a strategy vector for which $N(s) = G$.

2.2.3 Price of Stability of the Local Connection Game

Two types of graphs play a pivotal role in analyzing the Price of Stability: star graphs and the complete graph. A star graph first minimizes the number of built edges first to $n - 1$, and then, given this amount of edges, minimizes the distances (at most 2). The complete graph finds the absolute minimum in usage costs ($2n$), but the absolute maximum in construction costs ($\alpha n(n - 1)/2$).

Not surprisingly, then, a strategy that forms a star graph is optimal for large values of α, while a strategy resulting in the complete graph is optimal for low values of α. The tipping point is $\alpha = 2$. Intuitively, this is because for $\alpha < 2$, building an edge costs less than the minimal decrease in usage costs. For $\alpha > 2$, if the maximum distance between players is at most 2, then it is never profitable to build an edge, since the total decrease in usage costs is at most 2. This discussion is made precise in the following theorem:

2.2.1 Theorem. *A strategy vector s for $\mathrm{LCG}(n, \alpha)$ is optimal if and only if*

- $\alpha > 2$ *and $N(s)$ is a star graph, or*
- $\alpha < 2$ *and $N(s)$ is the complete graph, or*
- $\alpha = 2$ *and $N(s)$ has diameter of at most 2*

Proof. We first find a lower bound on the cost of s. The total construction cost is αm, where $m := |E(s)|$. There are precisely $2m$ ordered pairs of nodes

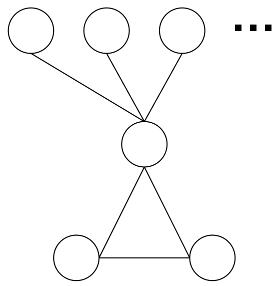

Figure 2.1: An example of a network that is neither a star nor the complete graph, but is still optimal if (and only if) $\alpha = 2$.

with distance 1 from eachother; these contribute $2m$ to the total usage cost. The other $n(n-1) - 2m$ pairs contribute at least $2 \cdot (n(n-1) - 2m)$ to the usage cost, since they are at least 2 edges removed from each other. So Cost(s) satisfies

$$\text{Cost}(s) \geq (\alpha - 2)m + 2n(n-1), \tag{2.2.2}$$

and equality is satisfied if and only if the diameter of $N(s)$ is at most 2.

For $\alpha > 2$ it follows that only graphs that minimize m to $n-1$ and have diameter at most 2 are optimal. Any such graph must be a star for the following reason. Consider two edges $\{u, v\}$ and $\{v, w\}$ in the graph. Any third edge must connect v with some vertex other than u and w; otherwise the diameter of the graph is at least 3, or $m > n-1$. It follows that v is the center in a star.

If $\alpha < 2$, the right side of (2.2.2) is minimized if m is as large as possible. The complete graph is the only graph satisfying the lower bound in this case.

If $\alpha = 2$, the right side of (2.2.2) reduces to $2n(n-1)$. A graph then satisfies the lower bound only if its diameter is at most 2. (See Figure 2.1.) □

Similarly to optimal strategies, when α is large, star graphs are generated by a Nash equilibrium, while the complete graph is generated a Nash equilibrium when α is small. The reasoning is about the same as in the case of optimal strategies, but the tipping point now lies at $\alpha = 1$. This reflects the fact that in the social cost function distances are counted twice, while a player is only interested in the distance to other players (and not from other players to him).

2.2.2 Theorem. *If $\alpha \geq 1$, then all star graphs are generated by a Nash equilibrium. If $\alpha \leq 1$, then the complete graph is generated by a Nash equilibrium.*

Proof. Suppose $\alpha \geq 1$. Any strategy that gererates a star graph is a Nash equilibrium. Indeed, no player has incentive to deviate by buying less edges, for then his usage cost would increase to infinity. The center player doesn't have to buy more edges because he is already connected to everyone. If another player buys $k \in \mathbb{Z}_{\geq 1}$ more edges, then his usage cost is reduced by k, but his construction cost is increased by $\alpha k \geq k$: the player doesn't profit by his deviation.

Suppose $\alpha \leq 1$. Any strategy that results in the complete graph is a Nash equilibrium. For if a player deviates by buying k less edges, he increases his usage cost by k (or infinity if $k = n-1$), while his construction cost decreases by only $\alpha k \leq k$. □

Theorems 2.2.1 and 2.2.2 allow for an easy computation of the Price of Stability of the Local Connection Game.

2.2.3 Theorem. *The Price of Stability of LocalConnectionGame*(n, α) *is equal to* 1 *if* $\alpha \notin (1, 2)$*, and lower than* $4/3$ *if* $\alpha \in (1, 2)$*. The bound of* $4/3$ *for* $\alpha \in (1, 2)$ *is strict.*

Proof. Suppose $\alpha \notin (1, 2)$. In this case, according to Theorems 2.2.1 and 2.2.2, the structure of optimal solutions and networks from certain Nash equilibria coincide. This means the Price of Stability is equal to 1.

Suppose $\alpha \in (1, 2)$. The complete graph is optimal in this case, while the star graph is generated by a Nash equilibrium. We find a strict upper bound for the Price of Stability by finding the worst ratio of costs between these two graphs. The ratio of costs for an instance LCG(n, α) is:

$$\frac{(\alpha - 2)(n - 1) + 2n(n - 1)}{(\alpha - 2)\frac{n(n-1)}{2} + 2n(n - 1)}.$$

This ratio increases as α decreases. The limit of the ratio as α approaches 1 is equal to:

$$\frac{2n(n - 1) - (n - 1)}{2n(n - 1) - \frac{n(n-1)}{2}} = \frac{4}{3} \cdot \frac{n^2 - \frac{3}{2}n + \frac{1}{2}}{n^2 - n} < \frac{4}{3}.$$

As n goes to infinity, the ratio approaches $4/3$. It follows that the bound is strict. □

2.2.4 Price of Anarchy of the Local Connection Game

We prove a couple of powerful results for the analysis of the Price of Anarchy of the Local Connection Game. In particular, the final result in this section, by Matúš Mihalák et al. [6], shows that the Price of Anarchy is at most 5 for ‘most’ instances, i.e. whenever $\alpha > 273n$. That’s because then the only Nash equilibria are trees. We will prove that trees have a Price of Anarchy of less than 5.

For very small α the analysis is finished quickly. The Price of Anarchy for the Local Connection Game when $\alpha < 1$ is simply calculated from the fact that the complete graph turns out to be the unique structure of the Nash equilibrium in this range of α.

2.2.4 Theorem. *If* $\alpha < 1$*,* $\rho(\text{LCG}(n, \alpha)) = 1$*.*

Proof. From Theorem 2.2.2 we know that the complete graph is generated by a Nash equilibrium if $\alpha \leq 1$. If $\alpha < 1$ the complete graph is in fact the only Nash equilibrium. For in any other graph there is a pair of players who are not connected to each other by an edge. Any one of those players can decrease his cost by buying the edge connecting the players; this decreases his usage cost by at least 1, while it costs him less than 1 to build the edge.

Since the complete graph also optimizes the social cost if $\alpha \leq 2$ (Theorem 2.2.1), this concludes the proof. □

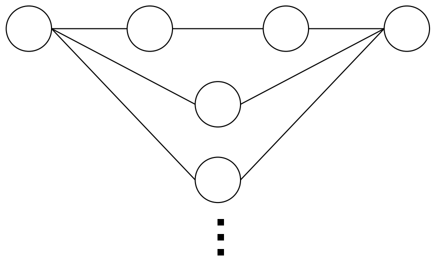

Figure 2.2: An example of a network where any associated Nash equilibrium has worse than optimal social cost. Any strategy that generates this network is a Nash equilibrium when $1 \leq \alpha \leq 4$. If on top of that $\alpha \neq 2$, then the social cost is worse than the optimal cost for $\text{LCG}(n, \alpha)$.

Finding bounds on the Price of Anarchy for larger values of α is more involved. Once $\alpha \geq 1$, instances can be found where a Nash equilibrium has worse than optimal social cost. For example, if $\alpha = 1$, the complete graph with one edge removed is a network for which any associated strategy vector is a Nash equilibrium. The social cost is 1 more than the optimal cost. Figure 2.2 shows an example for $1 \leq \alpha \leq 4$, $\alpha \neq 2$. Hence, the Price of Anarchy is greater than 1 if $\alpha \geq 1$.

Since the cost of a network is usually not only dependent on α, but also on the number of players n, it's interesting to find out what influence n has on the Price of Anarchy. One answer is that n can only have a very limited negative effect, in the sense that for fixed α, the Price of Anarchy of any instance is at most some fixed constant times $\sqrt{\alpha}$. We prove this in the following theorem.

2.2.5 Theorem. $\rho(\text{LCG}(n, \alpha)) \in \mathcal{O}(\sqrt{\alpha})$.[1]

The proof follows a two-step approach. First we find an upper bound on the cost of a Nash equilibrium as a function of the diameter of its resulting network. Then we find an upper bound on the diameter of said network. Together these bounds provide an upper bound for the Price of Anarchy of $\text{LCG}(n, \alpha)$.

The proof of the first part hinges on the observation that α is bounded by the diameter of the graph (times some constant), since it's a Nash equilibrium and no player can profit from deleting an edge.

2.2.6 Lemma. *Let s_{Nash} be a Nash equilibrium and s_{opt} be optimal for* $\text{LCG}(n, \alpha)$. *If $N(s_{\text{Nash}})$ has diameter d, then*

$$\frac{\text{Cost}(s_{\text{Nash}})}{\text{Cost}(s_{\text{opt}})} \leq 3d.$$

Proof. First we bound $\text{Cost}(s_{\text{opt}})$ from below. From (2.2.2) and $n > 1$ it follows that

$$\text{Cost}(s_{\text{opt}}) \geq \alpha(n-1) + n(n-1).$$

To find an upper bound for $\text{Cost}(s_{\text{Nash}})$, we divide the total construction cost in two parts: construction cost of cut edges[2] and construction cost of non-cut

[1]Recall that the notation $f(x) \in \mathcal{O}(g(x))$ means that there are constants c and M such that $f(x) \leq c \cdot g(x)$ for all $x \geq M$.

[2]A cut edge is an edge whose removal makes a graph disconnected

edges. Every distance is at most d, so the total usage cost is at most $dn(n-1)$. There are at most $n-1$ cut edges, so their total cost is at most $\alpha(n-1)$. We will prove that the costs of non-cut edges is bounded from above by $2dn(n-1)$. That means $\text{Cost}(s_{\text{Nash}})$ is bounded from above by $\alpha(n-1)+3dn(n-1)$. Since that is at most $3d$ times the lower bound on the optimal cost, this proves the theorem.

To find the bound for the costs of non-cut edges, pick a node u. We will bound the number $|p_u|$ of non-cut edges paid for by u by associating with each edge $e \in p_u$ the set V_e consisting of the nodes w where any shortest path between u and w must run through e. We will prove that $|V_e| \geq \alpha/2d$. Since there are only $n-1$ nodes besides u and the V_e are pairwise disjoint, it follows that $|p_u| \cdot \alpha/2d \leq n-1$, so $|p_u| \leq 2d(n-1)/\alpha$. Consequently, the construction cost for all non-cut edges is at most $2dn(n-1)$.

We will bound the increase in usage cost for u if the edge $e = \{u, v\} \in p_u$ is deleted. Let N_e denote the network $N(s_{\text{Nash}})$ with e deleted. The length of the shortest path P between u and v in N_e is at most $2d$. Indeed, let w be the first node in P that is in V_e, and w' be the node that precedes w in P. Since the shortest path between u and w' doesn't run along e even if it isn't deleted, the part of path P running from u to w' has length of at most d. The length between w and v is at most $d-1$, since the shortest path length between w and u if e is still there, is at most d. We conclude that the increase in distance from u to v is at most $2d-1$.

To reach any other $x \in V_e$ from u in N_e, we can first follow the path P to v and then find the shortest path to x. Any shortest path P' from v to x in $N(s_{\text{Nash}})$ still exists in N_e. If the length of the shortest path from u to x in $N(s_{\text{Nash}})$ is l, then the length of P' is $l-1$. It follows that the length of the shortest path from u to x in N_e is at most the length of the path $\langle P, P' \rangle$, which is at most $2d+l-1$. We conclude that the increase in distance from u to x is at most $2d-1$.

Consequently, deleting e would increase the usage costs of u by at most $2d|V_e|$. If u would delete e, it would save him α in cost. Since s_{Nash} is a Nash equilibrium, this should not be profitable. Therefore we must have $\alpha \leq 2d|V_e|$, and consequently $|V_e| \geq \alpha/2d$. □

2.2.7 Lemma. *Let s_{Nash} be a Nash equilibrium for* LCG(n, α)*. The diameter of $N(s_{\text{Nash}})$ is at most $2\sqrt{\alpha}$.*

Proof. We will prove by contraposition: if the diameter of a graph is more than $2\sqrt{\alpha}$, then it does not come from a Nash equilibrium. Let u and v be nodes for which the distance k of the shortest path P between them is more than $2\sqrt{\alpha}$.

If u builds the edge $\{u, v\}$[3], he would pay α. However, the distance between u and v would decrease by $k-1$. The distance between u and the node preceding v in P would decrease by $k-3$. Continuing on like this, we find that the total decrease in distance is at least

$$(k-1) + (k-3) + \ldots \geq \frac{k^2}{4}.$$

Since $k > 2\sqrt{\alpha}$, this means u improves his cost by building the edge $\{u, v\}$. Therefore, the graph isn't the result of a Nash equilibrium. □

[3]We assume $\alpha \geq 1$, so $\{u, v\}$ doesn't exist in the graph. Due to Theorem 2.2.4 we already know the Price of Anarchy for $\alpha < 1$.

Theorem 2.2.5 follows directly from the previous two Lemmas.

For large n (compared to α) the Price of Anarchy of any Local Connection Game-instance is actually only of the order of 1.

2.2.8 Theorem. *$\rho(\mathrm{LCG}(n, \alpha)) \in \mathcal{O}(1 + \alpha/\sqrt{n})$. In particular, $\rho(\mathrm{LCG}(n, \alpha)) \in \mathcal{O}(1)$ if $\alpha \in \mathcal{O}(\sqrt{n})$.*

The proof follows the same two-step approach as Theorem 2.2.5, with the exception that we use a bound different from the one found in Lemma 2.2.7. The theorem follows from Lemmas 2.2.6 and 2.2.9.

2.2.9 Lemma. *Let s_{Nash} be a Nash equilibrium for $\mathrm{LCG}(n, \alpha)$. The diameter of $N(s_{\mathrm{Nash}})$ is at most $9 + 4\alpha/\sqrt{n}$.*

Proof. Let d be the diameter of $N(s_{\mathrm{Nash}})$ and u, v be nodes in $N(s_{\mathrm{Nash}})$ with $\mathrm{dist}(u, v) = d > 1$. Let B_u be the set of nodes at most d' edges away from u, where $d' = \lfloor (d-1)/4 \rfloor$. First we find the following upper bound on $|B_u|$:

$$|B_u| \leq \alpha \cdot \frac{2}{d-1}. \tag{2.2.3}$$

Next we find a lower bound on $|B_u|$:

$$\frac{n(d'-1)}{2\alpha} \leq |B_u|. \tag{2.2.4}$$

With these two inequalities we find

$$\alpha^2 \geq \alpha \cdot \frac{(d-1)n(d'-1)}{4\alpha} \geq n(d'-1)^2.$$

Since $d \leq 4(d'+1) + 1$, it follows that $d \leq 4\alpha/\sqrt{n} + 9$, which concludes the proof of the lemma.

To find bound (2.2.3), suppose player v adds edge $\{u, v\}$ to $N(s_{\mathrm{Nash}})$. We will calculate how much this saves in usage costs for v. Consider a node $w \in B_u$. Before adding $\{u, v\}$ we must have $\mathrm{dist}(v, w) \geq d - d'$, since $\mathrm{dist}(u, v) = d$ is the shortest distance between u and v. After adding $\{u, v\}$, the distance decreases to at most $d' + 1$. It follows that v will save at least $(d - 2d' - 1)|B_u| \geq (d-1)|B_u|/2$ in usage costs. Since s_{Nash} is a Nash equilibrium, v should not be able to profit by adding any edge, so we must have $(d-1)|B_u|/2 \leq \alpha$.

To find bound (2.2.4), suppose player u adds edge $\{u, w\}$ to $N(s_{\mathrm{Nash}})$, where w is some node in B_u. We will calculate how much this saves in usage costs for u. Let A_w be the set of nodes t for which there is a shortest path from u to t that leaves the set B_u by passing through w. Note that if there is a shortest path leaving set B_u through w, it follows that $\mathrm{dist}(u, w) = d'$. So if $|A_w|$ is non-empty, player u saves at least $|A_w|(d'-1)$ in usage costs by buying edge $\{u, w\}$.

Let $w \in B_u$ be such that $|A_w|$ is above average. Since the union of all $A_{w'}$ contains all nodes outside of B_u, for node w the inequality $|A_w| \geq (n - |B_u|)/|B_u|$ holds. By choosing to connect with an edge to w, player u saves at least $(d'-1)(n - |B_u|)/|B_u|$ in usage costs. Since s_{Nash} is a Nash equilibrium, the savings cannot exceed α, i.e. $(d'-1)(n - |B_u|)/|B_u| \leq \alpha$. Rearranging yields the inequality

$$\frac{n(d'-1)}{(d'-1) + \alpha} \leq |B_u|.$$

We must have $\alpha \geq d > (d'-1)$, since otherwise u could profit by building edge $\{u,v\}$. Replacing $(d'-1)$ by α in the denominator of the above inequality gives the desired inequality (2.2.4). □

Where the previous two bounds were found by comparing the diameter of equilibrium networks with optimal networks, the next bound will be found by examining the structure of equilibrium networks. We will prove that, for sufficiently large α, all equilibrium networks are *trees* (connected graphs without cycles). That is a significant result, since trees are never far from optimal, as the following Theorem shows.

The proof uses the concept of a **subtree of a node** z in a tree T, i.e. a connected component in the subgraph of T generated by removing z and its adjoining edges. Note that there is a unique subtree $T_z(v)$ of z for each neighbor v of z, and $T_z(v) \cap T_z(w) = \emptyset$ for any two neighbors v,w of z.

2.2.10 Theorem. *Let s_{Nash} be a Nash equilibrium and s_{opt} an optimal strategy for* $\text{LCG}(n,\alpha)$. *If $N(s_{\text{Nash}})$ is a tree, then* $\text{Cost}(s_{\text{Nash}}) < 5 \cdot \text{Cost}(s_{\text{opt}})$.

Proof. We will first prove there is a **center node** $z \in N(s_{\text{Nash}})$, i.e. a node z for which the size of its largest subtree is at most $n/2$. For each node $v \in N(s_{\text{Nash}})$, let $T_v(w)$ denote the subtree of v corresponding to its neighbor w. Pick a node $v_0 \in N(s_{\text{Nash}})$. Consider a sequence $(v_i)_{i=0}^{\infty}$ of nodes generated by picking for each $i > 0$ a node $v_{i+1} \in N(s_{\text{Nash}})$ such that v_{i+1} is a neighbor of v_i and $|T_{v_i}(v_{i+1})|$ is maximal. We will prove by induction that some $v_k \in (v_i)_{i=0}^{\infty}$ is a center node.

Suppose v_i is not a center node. Then $|T_{v_i}(v_{i+1})| > n/2$. Note that each subtree of v_{i+1} except for $T_{v_{i+1}}(v_i)$ is a strict subset of $T_{v_i}(v_{i+1})$, so has cardinality at most $|T_{v_i}(v_{i+1})| - 1$. The subtree $T_{v_{i+1}}(v_i)$ is the complement of $T_{v_i}(v_{i+1})$ and therefore has cardinality at most $n/2$. This means that the cardinality of a maximal subtree of v_{i+1} is at most $\max\{|T_{v_i}(v_{i+1})| - 1, n/2\}$. By induction it follows that some v_i in the sequence must be a center node.

Let z be a center node of $N(s_{\text{Nash}})$ and suppose that the tree $N(s_{\text{Nash}})$ with z at its root has depth $d \geq 2$. Let v be a node at depth d and let T be the subtree of z for which $v \in T$. If v decides to buy the edge $\{v,z\}$, then it will decrease its distance to all nodes in the complement of T, since paths from v to those nodes must run through z. There are at least $n/2 - 1$ nodes in T's complement, since z is a center node, so $|T| \leq n/2$. Consequently, v saves at least $(d-1)(n/2-1)$. By the equilibrium constraint, we must have $(d-1)(n/2-1) \leq \alpha$.

Since $\text{diam}(N(s_{\text{Nash}}))$ is at most $2d$, by rearranging the above inequality we get $\text{diam}(N(s_{\text{Nash}})) \leq \frac{4\alpha}{n-2} + 2$. For the social cost of $N(s_{\text{Nash}})$ we note that there are $2(n-1)$ ordered pairs of nodes which are distance 1 apart from each other. The remaining $(n-2)(n-1)$ ordered pairs of nodes are at most $\text{diam}(N(s_{\text{Nash}}))$ removed from each other. Since $N(s_{\text{Nash}})$ is a tree, it contains precisely $n-1$ edges. This leads to the following inequality:

$$\begin{aligned}\text{Cost}(s_{\text{Nash}}) \leq \alpha(n-1) + 2(n-1) +& \\ (n-2)(n-1)\left(\frac{4\alpha}{n-2} + 2\right)& \\ = 5\alpha(n-1) + 2(n-1)^2.&\end{aligned}$$

By equation (2.2.2) we get

$$\frac{\mathrm{Cost}(s_{\mathrm{Nash}})}{\mathrm{Cost}(s_{\mathrm{opt}})} \leq \frac{5\alpha(n-1)+2(n-1)^2}{\alpha(n-1)+2(n-1)^2} < 5.$$

□

Proving that for sufficiently large α any equilibrium network is a tree, is done by looking in equilibrium networks at the average degree of a biconnected components in the network. A **biconnected component** of a graph G is a maximal subgraph H without any cut vertices. A **cut vertex** in a subgraph H is a vertex whose removal would make H disconnected. The **average degree** $\deg(H)$ of a subgraph is

$$\deg(H) = \frac{\sum_{v \in V(H)} \deg_H(v)}{|V(H)|},$$

where $\deg_H(v)$ denotes the degree of $v \in V(H)$ in H.

On the one hand, the average degree of a biconnected component is at least $2+\frac{1}{34}$ (Lemma 2.2.12). On the other hand, the average degree is *at most* $2+\frac{8n}{\alpha-n}$ (Lemma 2.2.13). This naturally leads to the conclusion that any equilibrium network for $\mathrm{LCG}(n,\alpha)$ with $\alpha > 273n$ cannot contain a biconnected component and is therefore a tree.

2.2.11 Theorem. *Let* $\mathrm{LCG}(n,\alpha)$ *be an instance of the Local Connection Game. If* $\alpha > 273n$*, then* $\rho(\mathrm{LCG}(n,\alpha)) \leq 5$*.*

Proof. Let s_{Nash} be a Nash strategy for $\mathrm{LCG}(n,\alpha)$. From Lemmas 2.2.12 and 2.2.13 we know that $N(s_{\mathrm{Nash}})$ cannot contain a biconnected component. Since all equilibrium networks are connected graphs, it follows that $N(s_{\mathrm{Nash}})$ is a tree. By Theorem 2.2.10, $\mathrm{Cost}(s_{\mathrm{Nash}}) < 5 \cdot \mathrm{Cost}(s_{\mathrm{opt}})$, where s_{opt} is an optimal strategy for $\mathrm{LCG}(n,\alpha)$. We conclude that $\rho(\mathrm{LCG}(n,\alpha)) \leq 5$. □

For the lower bound of $2+\frac{1}{34}$, we will prove that for any node c in a biconnected component H with $\deg_H(c) > 2$, there are at most $11 \cdot \deg_H(c)$ other nodes in H with degree 2. This means that $\deg(H)$ grows as the number of nodes in H with degree higher than 2 grows. Together with the fact that any biconnected component contains at least one node with degree higher than 2, this leads to the lower bound on $\deg(H)$.

2.2.12 Lemma. *Let* s_{Nash} *be a Nash equilibrium for* $\mathrm{LCG}(n,\alpha)$*, where* $\alpha > 19n$*. If* H *is a biconnected component of* $N(s_{\mathrm{Nash}})$*, then* $\deg(H) \geq 2+\frac{1}{34}$*.*

Proof. Let H be a biconnected component of $N(s_{\mathrm{Nash}})$. First note that there is at least one node $v \in V(H)$ with $\deg_H(v) \geq 3$ (Lemma A.4). Let D be the set containing these nodes. We can find a lower bound on $\deg(H)$ by finding an upper bound on the number of $v \in V(H)$ with $\deg_H(v) = 2$. (There are no nodes v in $V(H)$ with $\deg_H(v) < 2$, since then H would contain a cut vertex or be disconnected.)

The desired upper bound is found with the help of Lemma A.4. For each $d \in D$ define $V(d) = \{v \in V(H) : \mathrm{dist}(v,d) = \min_{d' \in D} \mathrm{dist}(v,d')\}$. To let the $V(d)$ form a partition of $V(H)$, if a node $v \in V(H)$ is closest to more than one

$d \in D$, we put v in only one of the $V(d)$. Since any $v \in V(H)$ is at most 11 edges removed from a $d \in D$, we have the following bound on $|V(d)|$:

$$|V(d)| \leq 1 + 11 \deg_H(d).$$

The desired lower bound follows:

$$\begin{aligned}
\deg(H) &= \frac{\sum_{d\in D} \deg_H(d) + 2 \cdot (|V(d)| - 1)}{|V(H)|} \\
&= \frac{2 \cdot \sum_{d\in D} |V(d)|}{|V(H)|} + \frac{\sum_{d\in D}(\deg_H(d) - 2)}{|V(H)|} \\
&\geq 2 + \frac{\sum_{d\in D}(\deg_H(d) - 2)}{|D| + \sum_{d\in D} 11 \deg_H(d)} \\
&\geq 2 + \frac{|D|}{|D| + 33|D|} \\
&= 2 + \frac{1}{34}.
\end{aligned}$$

□

For the upper bound of $2 + \frac{8n}{\alpha - n}$, instead of looking at the degree of nodes, we will consider the number of edges in a biconnected component H. We will find a particular spanning tree for H for which we can find a upper bound on the number of edges in H that are not in the spanning tree. Since the number of edges in the spanning tree is also bounded (by the number of nodes in H), this leads to our desired upper bound.

2.2.13 Lemma. *Let s_{Nash} be a Nash equilibrium for* $\text{LCG}(n, \alpha)$*, where $\alpha > n$. If H is a biconnected component of $N(s_{\text{Nash}})$, then* $\deg(H) \leq 2 + \frac{8n}{\alpha - n}$.

Proof. Let v_0 be a node (player) for which $u_{v_0}(s_{\text{Nash}})$[4] is minimal among the nodes $v \in V(N(s_{\text{Nash}}))$. Let T be a BFS-tree[5] of $N(s_{\text{Nash}})$ rooted at v_0. The graph $T' = T \cap H$ is a spanning tree of H. Indeed, any pair of nodes u,$v \in V(H)$ is connected with a path through H and a path through T. Since H is maximal, the path through T must also run completely through H. Consequently, T' is connected and spans H. This yields the following upper bound:

$$\deg(H) = \frac{2|E(T')| + 2|E(H) \setminus E(T')|}{|V(T')|} \leq 2 + \frac{2|E(H) \setminus E(T')|}{|V(T')|}.$$

To bound $|E(H) \setminus E(T')|$, we continue as follows. Let U be the set of nodes $v \in V(H)$ for which $(s_{\text{Nash}})_v \cap E(H) \setminus E(T') \neq \emptyset$. We prove the following:

1. $|U| = |E(H) \setminus E(T')|$
2. For each u,$v \in U$: $\text{dist}_{T'}(u, v) \geq \frac{\alpha - n}{2n}$,
3. $|U| \leq \frac{4n|V(T')|}{\alpha - n}$.

[4]Recall that $u_{v_0}(s_{\text{Nash}})$ denotes the usage cost of player v_0 given strategy vector s_{Nash}. See equation (2.2.1).

[5]We specifically use a Breadth First Search-tree, since we need to use its property that the path connecting the root v_0 of T with another node v is also a shortest path from v_0 to v in the network.

Part (2) implies part (3). This follows from a partition argument similar to the one in the proof of Lemma 2.2.12. For each $u \in U$ let $V(u) = \{v \in V(H) : \text{dist}(v,u) = \min_{u' \in U} \text{dist}(v,u')\}$. We stipulate that each $v \in V(H)$ can only be in one $V(u)$ and break ties arbitrarily. This means the $V(u)$ partition $V(H)$ and

$$|V(T')| = |V(H)| = \sum_{u \in U} |V(u)| \geq |U| \cdot \frac{\alpha - n}{4n}.$$

Parts (1) and (3) imply the desired bound:

$$\deg(H) \leq 2 + \frac{2|E(H) \setminus E(T')|}{|V(T')|} = 2 + \frac{2|U|}{|V(T')|} \leq 2 + \frac{8n}{\alpha - n}.$$

To prove part (1), assume player $u \in U$ buys $k \in \mathbb{Z}_{>1}$ edges in $E(H) \setminus E(T')$ and $l \in \mathbb{Z}_{\geq 0}$ other edges. Player u can improve its cost by not buying the edges in $E(H) \setminus E(T')$ and buying an edge to v_0 instead, constradicting that s_{Nash} is a Nash equilibrium. Denote the new strategy vector by s. Then

$$\begin{aligned}\text{Cost}_u(s) \leq u_{v_0}(s_{\text{Nash}}) + n + l \cdot \alpha &\leq u_u(s_{\text{Nash}}) + n + l \cdot \alpha \\ &< u_u(s_{\text{Nash}}) + (k + l)\alpha \\ &= \text{Cost}_u(s_{\text{Nash}}),\end{aligned}$$

where the second inequality holds because $u_{v_0}(s_{\text{Nash}})$ is minimal and the third holds because $\alpha > n$. It follows that each $u \in U$ buys exactly one edge in $E(H) \setminus E(T')$, so $|U| = |E(H) \setminus E(T')|$.

To prove part (2), let u, $v \in U$, $u \neq v$, be such that $\text{dist}_{T'}(u,v) < \frac{\alpha - n}{2n}$. Let $P = \{x_1, \ldots, x_k\}$ be the shortest path in T' connecting u and v (so $x_1 = u$ and $x_k = v$). Let u', $v' \in V(H)$ be such that $\{u,u'\}$, $\{v,v'\} \in E(H) \setminus E(T')$.

We first prove that v_0,u',$v' \notin P$. By Lemma A.1, both u' and v' are not a descendant in T' of any $x_i \in P$. Otherwise the u'-x_i-path in T', the x_i-u-path in T' and $\{u,u'\}$ (and similarly for v') form a cycle in $N(s_{\text{Nash}})$ of length at most[6] $2(d_{T'}(u,v) + 1) < \frac{\alpha}{n} + 1$. Consequently, u' and v' are not in P and $P' = \{x_0, \ldots, x_{k+1}\}$, where $x_0 = u'$ and $x_{k+1} = v'$, is a path in H. Lemma A.1 also implies that $\text{dist}(u,v_0) > \text{dist}(u,v)$, so $\text{dist}(u,v_0) \geq \frac{\alpha - n}{2n}$ (and similarly for v). Therefore $v_0 \notin P$.

Let $x_i \in P$ be such that x_i buys both $\{x_i, x_{i+1}\}$ and $\{x_i, x_{i-1}\}$. Such a node exists, since x_1 buys $\{u', x_1\}$ and x_k buys $\{x_k, v'\}$. Let x_i unilaterally deviate from s_{Nash} as follows: x_i no longer buys the two edges in P, but buys $\{x_i, v_0\}$ instead. Denote the new strategy vector by s. We show that $\text{Cost}_{x_i}(s) < \text{Cost}_{x_i}(s_{\text{Nash}})$. This leads to a contradiction with s_{Nash} being a Nash equilibrium and thereby proves the theorem.

In s, $c_{x_i}(s) = c_{x_i}(s_{\text{Nash}}) - \alpha$. In the remaining we show that $u_{x_i}(s) < u_{x_i}(s_{\text{Nash}}) + \alpha$. This proves that $\text{Cost}_{x_i}(s) = c_{x_i}(s) + u_{x_i}(s) < \text{Cost}_{x_i}(s_{\text{Nash}})$. Note that $u_{x_i}(s) \leq u_{v_0}(s) + n$, since there are n nodes in $N(s_{\text{Nash}})$. It remains for us to show that $u_{v_0}(s) < u_{v_0}(s_{\text{Nash}}) + \alpha - n$.

The only nodes that might have increased distance to v_0 in $N(s)$ compared with $N(s_{\text{Nash}})$, are the nodes in P and their descendants in T. Let w be such a node. Let x_j be the ancestor of w in T closest to w among the nodes in P. Since T is a BFS-tree, we have

$$\text{dist}_{N(s_{\text{Nash}})}(v_0, w) = \text{dist}_T(v_0, x_j) + \text{dist}_T(x_j, w).$$

[6] T' is a BFS-tree, so $\text{dist}_{T'}(u', x_i) \leq \text{dist}_{T'}(u, x_i) + 1$.

Suppose $j = i$. The path via T is still present in the new network $N(s)$, so $\mathrm{dist}_{N(s)}(v_0, w) = \mathrm{dist}_{N(s_{\mathrm{Nash}})}(v_0, w)$. Suppose $j < i$. Since x_0 is not a descendant from any node in P and the path $\{x_0, \ldots, x_j\}$ remains intact in $N(s)$, we have

$$\begin{aligned}\mathrm{dist}_{N(s)}(v_0, w) &\le \mathrm{dist}_{N(s)}(v_0, x_0) + \mathrm{dist}_{N(s)}(x_0, x_j) + \mathrm{dist}_{N(s)}(x_j, w)\\ &= \mathrm{dist}_T(v_0, x_0) + \mathrm{dist}_{P'}(x_0, x_j) + \mathrm{dist}_T(x_j, w).\end{aligned}$$

So the increase in distance from v_0 to w is

$$\begin{aligned}\mathrm{dist}_{N(s)}(v_0, w) - \mathrm{dist}_{N(s_{\mathrm{Nash}})}(v_0, w) &\le \mathrm{dist}_T(v_0, x_0) + \mathrm{dist}_{P'}(x_0, x_j)\\ &\qquad - \mathrm{dist}_T(v_0, x_j)\\ &\le 2 \cdot \mathrm{dist}_{P'}(x_0, x_j)\\ &\le 2 \cdot \mathrm{dist}_{T'}(u, v) < \frac{\alpha - n}{n}.\end{aligned}$$

A similar argument applies when $j > i$.

We conclude that the total increase in usage cost is less than $n \cdot \frac{\alpha - n}{n}$, so $u_{v_0}(s) < u_{v_0}(s_{\mathrm{Nash}}) + \alpha - n$, which we wanted to prove. □

2.3 Global Connection Game

2.3.1 Introduction

In the Global Connection Game players want to build a path from a source to a sink while keeping construction costs as low as possible. If multiple players decide to build an edge, they share the costs equally. This type of cost allocation called *fair cost allocation* is often used in network design. It follows from the Shapley value, a solution concept in cooperative game theory [8].

Two examples in this section yield strict upper bounds on both the Price of Anarchy and Stability. Most interesting properties of the Global Connection Game follow from the fact that it is a potential game. See Chapter 3 for more details.

The Global Connection Game was introduced by Elliot Anshelevich et al. in [2], where it was called the *Shapley-value cost sharing game*. We call it the Global Connection Game to contrast with the Local Connection Game from Section 2.2. In the Local Connection Game, players have no influence on 'distant' parts of the network, i.e. on connections between pairs of users that don't include themselves. In the Global Connection Game, the users influence the structure of a network across multiple nodes.

2.3.2 The model

Let G be a directed graph where each edge $e \in E(G)$ has a fixed **cost** $c_e \in \mathbb{R}_{\ge 0}$. An **instance** $\mathrm{GCG}(G)$ of the Global Connection Game is a finite strategic game with player set $A = \{1, \ldots, n\}$. Each player i is assigned a pair of nodes $\{s_i, t_i\} \subseteq V(G)$ — his **source node** and **sink node**. The strategy set $\mathcal{P}_i$ of player i is the set of paths from s_i to t_i. A strategy vector $s \in \mathcal{S} = \Pi_{i=1}^n \mathcal{P}_i$ generates a network $N(s)$ with $V(N(s)) = V(G)$ and $E(N(s)) = \cup_i P_i$.

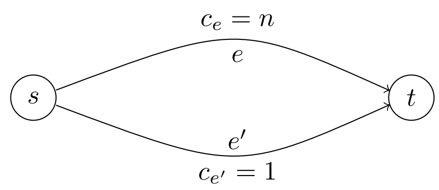

Figure 2.3: A Global Connection Game with the worst possible Price of Anarchy, n, and the best possible Price of Stability, 1. See Example 2.3.1.

The cost that a player incurs by including edge e in his chosen path depends on how many other players chose e. In our model, the cost of building an edge is divided evenly among the players, i.e. if k players chose e in their paths, the cost per player is c_e/k. We let $\text{Cost}_e(s)$ denote the construction cost per player of edge e, given a strategy vector s. The cost for a player i is the sum of his edge costs:

$$\text{Cost}_i(s) = \sum_{e \in s_i} \text{Cost}_e(s) = \sum_{e \in s_i} \frac{c_e}{n_e(s)},$$

where $n_e(s)$ is the number of players with e in their chosen path. The social cost of a strategy s is simply the cost of all edges built in the network $N(s)$:

$$\text{Cost}(s) = \sum_{e \in N(s)} c_e = \sum_{i=1}^{n} \text{Cost}_i(s).$$

Other concepts such as the Price of Anarchy are defined as in Appendix B.

2.3.3 Price of Anarchy and Stability of the Global Connection Game

Strict bounds on the Price of Anarchy and Stability for the Global Connection Game are quickly found in two simple examples.

2.3.1 Example. Consider a graph with two nodes s and t and two edges e and e' from s to t with costs n and 1, respectively. For each player i, $s_i = s$ and $t_i = t$. (See Figure 2.3.)

The social cost is 1, n or $n+1$, depending on which edges are built. The social optimum is therefore equal to 1. Suppose k players buy edge e. If $0 < k < n$, any one of those players can decrease his cost by buying edge e' instead, so this is not a Nash equilibrium. If $k = n$, the cost for a player doesn't decrease if he buys edge e' instead, and if $k = 0$, the cost for a player increases if he buys edge e instead of e'.

These are the only two Nash equilibria. They have costs n and 1, respectively, so the Price of Anarchy of this example is n while the Price of Stability is 1.

2.3.2 Example. Consider a graph with nodes t, v, and s_i for each player i. There is an edge from v to t with cost $1+\epsilon$ $(0 < \epsilon < n)$, for each i an edge from s_i to v with zero cost, and for each i an edge from s_i to t with cost $1/i$. The sink node for each player is t. (See Figure 2.4.)

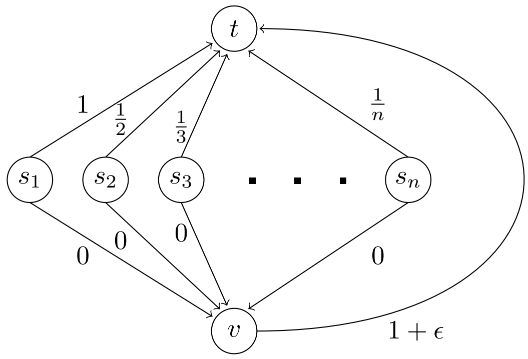

Figure 2.4: A Global Connection Game with the worst possible Price of Stability, $\mathcal{H}_n$. See Example 2.3.2.

The optimal cost is $1+\epsilon$, when all players choose the route via v. Any other strategy will result in a cost strictly higher than $1+\epsilon$. This is, however, not a Nash equilibrium.

In fact, there is only one Nash equilibrium: the strategy where player i chooses the edge from s_i to t, for each i. Its cost is $\mathcal{H}_n$, the n-th harmonic number. If $k > 0$ players choose the route via v, any one of those players i with $i \geq k$ can reduce his cost from $(1+\epsilon)/k$ to at most $1/k$ by choosing the route directly to t instead.

In this example, therefore, both the Price of Stability and the Price of Anarchy are (roughly) equal to $\mathcal{H}_n$.

Both these examples provide worst-case scenarios for the Price of Anarchy and Stability: Example 2.3.1 for the Price of Anarchy and Example 2.3.2 for the Price of Stability. Proving the upper bound for the Price of Anarchy is straightforward.

2.3.3 Theorem. *The Price of Anarchy of a Global Connection Game is never higher than n, the number of players. This is a strict upper bound.*

Proof. The fact that this is the smallest upper bound follows from Example 2.3.1. Let s be a strategy with $\mathrm{Cost}(s) > n \cdot \mathrm{Cost}(s_{\mathrm{opt}})$, where s_{opt} is a strategy with optimal social cost. We will prove that s is not a Nash equilibrium.

Since the social cost function is the sum of the players' costs, there is a player i with $\mathrm{Cost}_i(s) > \mathrm{Cost}(s_{\mathrm{opt}})$. However, the path $(s_{\mathrm{opt}})_i$ satisfies $\mathrm{Cost}_i(s') \leq \mathrm{Cost}(s_{\mathrm{opt}})$ for any strategy vector s'. This is because the social cost function is also the sum of the maximum costs for each edge, and $(s_{\mathrm{opt}})_i$ is a subset of the edges in $N(s_{\mathrm{opt}})$. So if player i changes his strategy to $(s_{\mathrm{opt}})_i$, he decreases his cost to at most $\mathrm{Cost}(s_{\mathrm{opt}})$. Hence, s is not a Nash equilibrium. □

The Price of Stability will never exceed $\mathcal{H}_n$. The bound follows directly from the properties of *potential games*, as the Global Connection Game turns out to be a game of this type. The class of potential games is discussed in Chapter 3.

2.4 Facility Location Game

2.4.1 Introduction

The games we have discussed so far all lack interaction with customers. For example, the players in the Local Connection Game only strive to build a well connected yet cheap network. Of course, the customer base is implied: once the network is created, the Internet Service Providers can sell their services on the network to the internet users. Nevertheless, we would like to see what happens to the Price of Anarchy when we have a game with an explicit pricing component.

The game we study in this section is the Facility Location Game, introduced by Adrian Vetta in [18]. In this game, service providers compete over a set of customers. The service providers each have a few possible locations where they can establish their facility. Different locations have different costs for connecting to the various customers. The profits of a provider come from the customers' payments.

An aspect of the Facility Location Game that distinguishes the game from the others in this thesis, is the social utility function. In the other games the social utility was only dependent on the utility (cost) of each player. In the Facility Location Game, in contrast, the utility that is created by connections is also taken into account. Specifically, each customer has a certain value for each service. The welfare created by a connection is this value minus the building cost.

The Facility Location Game is an instance of a broader class of games called *Utility Games*. In Section 2.5 we prove that the Facility Location Game is a *Basic, Monotone Utility Game*. A consequence is that each instance of the Facility Location Game has Price of Anarchy at most 2 and Price of Stability equal to 1.

2.4.2 The model

Consider a complete bipartite graph $G = (F \cup C, E)$. A node $f \in F$ is called a **facility location**, a node $c \in C$ is called a **customer**. An **instance** FLG(G) of the Facility Location Game is a finite strategic game. The players $A = \{1, \dots, n\}$ are called **service providers**. The strategy set of a service provider i is denoted by F_i. The F_i are pairwise disjoint and $\cup_{i=1}^{n} F_i = F$. Each customer $c \in C$ has an associated number $v(c) \in \mathbb{R}_{\geq 0}$, which represents the **value** it puts on being serviced. Each edge $e = \{f, c\}$, where $f \in F$, $c \in C$, has an associated number $c(e) = c(f, c) \in \mathbb{R}_{\geq 0}$, which represents the **cost to service customer** c from facility location f.

To define the utility functions, we need to describe how customers are assigned to facilities and how prices are set. Given a strategy vector s, each customer c chooses the facility s_i that can service for the lowest cost $c(s_i, c)$. The price $p(i, c)$ that service provider i charges customer c is the cost $c(s_k, c)$ of the second cheapest service available to c, given the locations chosen by the players (so $p(i, c) = \min_{k \neq i} c(s_k, c)$). Intuitively, this is the highest possible price that i can offer c without giving incentive to c to run to a competitor.

If customer c is serviced from location s_i by service provider i, then the net value generated for customer c is $v(c) - p(i, c)$. The net value generated for service provider i is $p(i,c) - c(s_i, c)$. This leads to the following definitions of utility in the instance FLG(G). The utility function V_i of player i is

$$V_i(s) = \sum \{p(i,c) - c(s_i, c) : c(s_i, c) = \min_{k=1}^{n} c(s_k, c)\}, \tag{2.4.1}$$

where $p(i,c) = \min_{k \neq i} c(s_k, c)$. The social utility function V of G is

$$V(s) = \sum_{c \in C} v(c) - \min_{i=1}^{n} c(s_i, c),$$

which is simply the sum of the generated values for the customers *and* the service providers. Note that the prices $p(i,c)$ have no influence on the social utility function.

We will make the natural assumption that a customer c only chooses to be serviced by i if $v(c) \geq p(i,c)$, while service provider i only provides service to c from s_i if $p(i,c) \geq c(s_i, c)$. Furthermore, we assume that $v(c) \geq c(i,c)$ for all $i \in F$, $c \in C$. This requires no loss of generality, since if $v(c) < c(i,c)$, we can decrease $c(i,c)$ to the same value as $v(c)$ without changing the values generated for c and i in any assignment. Indeed, if $v(c) < c(i,c)$, then customer c will not choose to be serviced from i, so no value is generated between i and c. If $v(c) = c(i,c)$, then c could be serviced from i, but this also generated no value for both i and c.

Nash equilibrium, optimality, and the Price of Anarchy and Stability are defined as in Appendix B.

2.4.3 Properties of the FLG

Every instance of the FLG has a Nash equilibrium and a Price of Stability equal to 1. These are some of the consequences of the fact that the FLG is part of a class of games known as potential games. This class of games is discussed in Chapter 3. The FLG is also part of another class of games, the utility games.

2.5 Utility Games

2.5.1 Introduction

The class of Utility Games is a generalization of the Facility Location Game from Section 2.4. In a Utility Game, players can choose one out of a set of locations. The combined set of locations that the players choose produces a certain amount of 'social welfare', determined by the *social welfare function* associated with the game. The social welfare function is like a social utility function. However, in contrast to the other games from this thesis, the social welfare generated may be greater than just the sum of the profits of each player.

The social welfare function of a Utility Game must satisfy a few conditions. These conditions are meant to make Utility Games an economically realistic model. For instance, the required *submodularity* property means that the social welfare has *diminishing marginal utility*, i.e. the benefit of adding a location decreases (or at least does not increase) as the existing set of locations grows larger.

A subclass of Utility Games called *Basic Utility Games* are potential games. In Chapter 3 we prove this fact and several of its consequences.

2.5.2 Model

Let G be a finite strategic game with player set $A = \{1, \ldots, n\}$. An element of the strategy set A_i of player i is called a **location**. Let $A = \cup_{i=1}^{n} A_i$ and $S = \Pi_{i=1}^{n} A_i$. The utility function of player i is denoted by α_i. We are given a function $V : \mathcal{P}(A) \to \mathbb{R}$ called the **social welfare function**. For $s \in S$ we take $V(s)$ to mean $V(\cup_{i=1}^{n} \{s_i\})$.

We call G a Utility Game if (1) the marginal benefit of adding a location doesn't increase as the number of existing locations gets larger, (2) the total welfare of the players doesn't exceed the social welfare and (3) each player's welfare is at least as large as his contribution to the social welfare. This is formalized in the following definition.

2.5.1 Definition. A finite strategic game G with n players and social welfare function V is called a **Utility Game** if it satisfies the following three properties:

1. V is **submodular**: for each pair of sets S, S' with $S \subseteq S' \subseteq A$ and each $u \in A$ the following inequality holds:

$$V(S \cup \{u\}) - V(S) \geq V(S' \cup \{u\}) - V(S').$$

2. $\sum_{i=1}^{n} \alpha_i(s) \leq V(s)$ for each strategy vector s.

3. $\alpha_i(s) \geq V(s) - V(s \setminus \{s_i\})$ for each player i and strategy vector s.

If, furthermore, G satisfies

3′. $\alpha_i(s) = V(s) - V(s \setminus \{s_i\})$ for each player i and strategy vector s,

then G is called a **Basic Utility Game**. If a Utility Game satisfies

4. $V(S) \leq V(S')$ for all $S \subseteq S' \subseteq A$,

then G is called a **Monotone Utility Game**.

2.5.3 Facility Location Game as a Utility Game

The Facility Location Game satisfies the whole range of properties from Definition 2.5.1. Before we prove this, however, we must slightly change the definition of the Facility Location Game, so that it can be incorporated in the context of this section. We only need to extend the social utility function V of an instance $I = \text{FLG}(G)$ so that it is defined on any subset $S \subseteq F$. For this we take V to be the function $V : \mathcal{P}(F) \to \mathbb{R}$ defined by

$$V(S) = \sum_{c \in C} v(c) - \min_{u \in S} c(u, c).$$

2.5.2 Theorem. *Every instance* FLG(G) *of the Facility Location Game is a Basic, Monotone Utility Game.*

Proof. We check the properties in order of Definition 2.5.1.

1. Let S, S' be sets such that $S \subseteq S' \subseteq F$. Suppose location u is added to S or S'. We will examine the change in social welfare due to a customer c. This customer will only change his location of his value will be improved.

   If c doesn't change to location u from its location in S', then the change in welfare from S' to $S' \cup \{u\}$ due to c is 0. If c does change its connection to u from its connection in S', then it also changes to u from its connection in S. This is because $\min_{u \in S} c(u,c) \geq \min_{u \in S'} c(u,c)$. From this inequality it also follows that the change in welfare due to c is bigger when starting with S than with S':

   $$c(u,c) - \min_{v \in S} c(v,c) \geq c(u,c) - \min_{v \in S'} c(v,c). \tag{2.5.1}$$

   It follows that if we sum for all $c \in C$, the desired inequality holds.

2. The social welfare function is the sum of the players' welfare and the clients' welfare. Recall that we assume that each player's and client's welfare is nonnegative, so this proves the desired inequality.

3′. Let s be a strategy vector. If facility s_i is removed from s, then the only change in social welfare is due to the customers who were connected to s_i. These will now be serviced by the cheapest facility in $s \setminus \{s_i\}$. Let c be such a customer. The change in welfare due to c is

   $$v(c) - \min_{k=1}^{n} c(s_k, c) - (v(c) - \min_{k \neq i} c(s_k, c)) = \min_{k \neq i} c(s_k, c) - \min_{k=1}^{n} c(s_k, c).$$

   The sum of these values over all c connected to s_i is exactly player i's welfare (2.4.1).

4. Suppose $S \subseteq S' \subseteq F$, then $\min_{u \in S} c(u,c) \geq \min_{u \in S'} c(u,c)$ for each customer c, so $V(S) \leq V(S')$.

□

2.5.4 Price of Anarchy and Stability of Utility Games

The social welfare function V of a Basic Utility Game has the nice property that $-V$ is a 'potential function' (Theorem 3.3.2). Since the minimum of a potential function is a Nash equilibrium (Theorem 3.4.1), a direct consequence is that the Price of Stability of all Basic Utility Games is 1 (Theorem 3.5.3). For more details, see Chapter 3.

For Monotone Utility Games we can deduce the following positive result for their Price of Anarchy, provided a Nash equilibrium exists.

2.5.3 Theorem. *Let G be a Monotone Utility Game. If there exists a Nash equilibrium for G, then $\rho(G) \leq 2$.*

Proof. Let s_{Nash} be a Nash equilibrium and s_{opt} be an optimal strategy vector for G. We will prove that $V(s_{\text{opt}}) \leq 2V(s_{\text{Nash}})$ by finding a series of inequalities that follow directly from the properties (1)–(4) in Definition 2.5.1. Here we use the notation s^i_{opt} for the i-th coordinate of s_{opt} and $s^{\le i}_{\text{opt}}$ for the set $\cup_{k=1}^{i} s^k_{\text{opt}}$ (and $s^{\le 0}_{\text{opt}} := \emptyset$).

$$
\begin{aligned}
V(s_{\text{opt}}) - V(s_{\text{Nash}}) &\overset{(4)}{\le} V(s_{\text{Nash}} \cup s_{\text{opt}}) - V(s_{\text{Nash}}) \\
&= \sum_{i=1}^{n} V(s_{\text{Nash}} \cup s^{\le i}_{\text{opt}}) - V(s_{\text{Nash}} \cup (s^{\le i-1}_{\text{opt}})) \\
&\overset{(1)}{\le} \sum_{i=1}^{n} V(s_{\text{Nash}} + s^i_{\text{opt}} - s^i_{\text{Nash}}) - V(s_{\text{Nash}} - s^i_{\text{Nash}}) \qquad (2.5.2) \\
&\overset{(3)}{\le} \sum_{i=1}^{n} \alpha_i(s_{\text{Nash}} + s^i_{\text{opt}} - s^i_{\text{Nash}}) \\
&\overset{\text{Nash}}{\le} \sum_{i=0}^{n} \alpha_i(s_{\text{Nash}}) \\
&\overset{(2)}{\le} V(s_{\text{Nash}}).
\end{aligned}
$$

□

Chapter 3

Potential Games

3.1 Introduction

This chapter studies games that admit a 'potential function'—a single function that tracks changes in utility as players change their strategy. The mere existence of a potential function has strong consequences for existence of Nash equilibria, the Price of Stability and convergence to Nash equilibria, as we will see in Section 3.4. Several games we have discussed in this thesis are potential games. We will prove in Section 3.3 that every instance of the Global Connection Game (Global Connection Game) (see Section 2.3) and every Basic Utility Game (see Section 2.5) is a potential game. Every instance of the Atomic Rounting Game (see Chapter 1) is also a Potential Game, provided that each player has the same traffic rate or each edge cost function is affine.

The first extensive study of potential games and its applications was done in [7]. Most of the theorems on potential games in this chapter come from that article. For Theorem 3.2.3 we give an alternative, constructive proof.

3.2 Definition and a characterization

3.2.1 Definition. Let $G = (A, S, U)$ be a finite strategic game. A function $\Phi : S \to \mathbb{R}$ is called an **(exact) potential function** on G if it satisfies the following condition for each player $i \in A$:

$$\Phi(s) - \Phi(s') = u_i(s') - u_i(s) \tag{3.2.1}$$

for each pair of strategies s, $s' \in S$ that differ only on the i-th coordinate.

A game G for which a potential function exists is called a **potential game**.

A potential function tracks the savings a player incurs if he unilaterally changes strategies. Any two potential functions on a game G differ by only a constant, as a simple calculation shows. Indeed, if Φ, Ψ are potential functions and s, s' strategies on G, then $\Phi(s) - \Psi(s) = \Phi(s') - \Psi(s')$.

Potential functions satisfy a kind of 'law of conservation of utility', by which we mean the following. Suppose you sum the changes in utility experienced by the players in each step of a **strategy cycle** $C = \langle s^1, \ldots, s^k \rangle$, i.e. a sequence of strategies where s^i and s^{i+1} differ on only one coordinate for each $1 \leq i < k$

and $s^k = s^1$, then the sum will add up to 0. In fact, this property characterizes potential games, as we will now prove.

Let A_i denote the strategy set of player i. Identify the strategies in A_i with the numbers in the set $N_i = \{1, \ldots, |A_i|\}$. Consider the set $N = \Pi_{i=1}^n N_i$. We call N a **Nash grid** of G. In the remaining we view each element $a \in N$ as a strategy in S^G. For each pair a,$b \in N$ that differ on precisely one coordinate, we let $p(a, b)$ denote the player that changes his strategy between a and b and define the **utility change** $u(a, b)$ from a to b as

$$u(a, b) = u_{p(a,b)}(b) - u_{p(a,b)}(a).$$

Note that $u(b, a) = -u(a, b)$. A strategy cycle in N is defined as above. A strategy path is defined as a strategy cycle where the end points do not necessarily have to meet. The utility change $u(P)$ of a strategy path $P = \langle a^1, \ldots, a^k\rangle$ in N is

$$u(P) = \sum_{i=1}^{k-1} u(a^i, a^{i+1}).$$

Sometimes it is useful to consider the curve $L(P)$ that P follows through $\mathbb{R}^n$. The curve is defined as $L(P) = \cup_{i=1}^{k-1}[a^i, a^{i+1}]$, where $[a^i, a^{i+1}]$ is the closed line segment in $\mathbb{R}^n$ joining a^i and a^{i+1}. Note that $u(P) = u(P')$ whenever $L(P) = L(P')$ and P and P' have the same endpoints. We say that P and P' are **equivalent**. If the endpoints are flipped, $u(P) = -u(P')$.

3.2.2 Theorem. *A finite game G is a potential game if and only if, for all cycles C in a Nash grid N of G, $u(C) = 0$.*

Proof. Suppose G is a potential game with potential funtion Φ. Let $C = \langle a^1, \ldots, a^k\rangle$ be a strategy cycle in N. Then the following holds:

$$u(C) = \sum_{i=1}^{k-1} u(a^i, a^{i+1}) = \sum_{i=1}^{k-1} \Phi(a^i) - \Phi(a^{i+1}) = \Phi(a^1) - \Phi(a^k) = 0.$$

Suppose all strategy cycles in N have utility change 0. We construct a potential function Φ for G as follows. Let $e = (1, \ldots, 1)$ and set $\Phi(e) = 0$. For each $a \in N$, set $\Phi(a) = -u(P)$, where P is a strategy path from e to a. To prove that this is well-defined, consider two paths P_1, P_2 from e to a. Let C be the cycle $C = \langle P_1, P_2^{-1}\rangle$, where P_2^{-1} is the path P_2 traversed in the opposite direction (from a to e). By assumption, $u(C) = 0$. Since $u(C) = u(P_1)+u(P_2^{-1})$ and $u(P_2^{-1}) = -u(P_2)$, it follows that $u(P_1) = u(P_2)$.

The constructed function Φ is a potential function. Indeed, consider strategies a,$b \in N$ that differ on precisely one coordinate. Let P be a path from e to a. Then $P' = \langle P, b\rangle$ is a path from e to b and

$$\Phi(a) - \Phi(b) = u(P') - u(P) = u(a, b).$$

So the condition (3.2.1) holds for each pair a,$b \in N$ that differ on only one coordinate. □

We can prove an even stronger characterization than Theorem 3.2.2. We call a strategy cycle $C = \langle a_1, \ldots, a_k\rangle$ in a Nash grid N **elementary** if $k = 5$. Note that at most two players change strategies in an elementary strategy cycle.

3.2.3 Theorem. *A finite game G is a potential game if and only if, for all elementary strategy cycles C in a Nash grid N of G, $u(C) = 0$.*

Proof. If G is a potential game, Theorem 3.2.2 implies $u(C) = 0$ for each elementary strategy cycle. Conversely, suppose all elementary strategy cycles in N have utility change 0. We will prove that any strategy cycle has utility change 0. By Theorem 3.2.2, this proves that G is a potential game.

Let $C = \langle a^1, \ldots, a^k \rangle$ be a strategy cycle in N. Applying Lemma 3.2.4 recursively, we 'collapse' C to a strategy cycle C' for which $u(C') = u(C)$ and $C' \subseteq \{a \in N : a_i = 1 \text{ for each } 2 \leq i \leq n\}$. Since C' is a cycle that runs along only one line in N, we must have $u(C') = 0$. This proves that $u(C) = 0$. □

3.2.4 Lemma. *Let N be a Nash grid of a finite game G, C a strategy cycle in N and $i \in \{1, \ldots, n\}$. Suppose all elementary strategy cycles in N have utility change 0. There is a strategy cycle C' in N such that $u(C') = u(C)$ and $C' \subseteq \{a \in N : a_i = 1\}$.*

Proof. We prove the statement for $i = n$. The general case is proven similarly. Let $C = \langle a^1, \ldots, a^k \rangle$ be a strategy cycle in N. For each $1 \leq j < k$ define the vector $b^j = (a^j_1, \ldots, a^j_{n-1})$. Suppose a^j is such that $p(a^j, a^{j+1}) \neq n$. The assumption that elementary cycles have zero utility change implies that the path

$$\langle a^j = (b^j, a^j_n), (b^j, a^j_n - 1), (b^{j+1}, a^j_n - 1), (b^{j+1}, a^j_n) = a^{j+1} \rangle$$

has the same utility change as the path $\langle a^j, a^{j+1} \rangle$. By induction

$$u\left(a^j, a^{j+1}\right) = u\left(\langle a^j, (b^j, 1), (b^{j+1}, 1), a^{j+1} \rangle\right).$$

We use this fact to replace C with another strategy cycle with equal utility change. For each $1 \leq j < k$ such that $p(a^j, a^{j+1}) \neq n$, replace the subpath $\langle a^j, a^{j+1} \rangle$ of C by the subpath

$$\langle a^j, (b^j, 1), (b^{j+1}, 1), a^{j+1} \rangle,$$

and name the resulting strategy cycle $\tilde{C}$. This cycle satisfies $u(C) = u(\tilde{C})$.

However, $\tilde{C}$ may run along points $a \in N$ with $a_n \neq 1$, because we only collapsed the segments in C where $p(a^j, a^{j+1}) \neq n$. We will show that the segments of $\tilde{C}$ that do run along such points, cancel each other out.

Let $1 \leq j < k$ be such that $p(a^j, a^{j+1}) \neq n$ and let a^l be the next point on C for which $p(a^l, a^{l+1}) \neq n$ (possibly $l = j$). In $\tilde{C}$, $(b^{j+1}, 1)$ is followed by a^{j+1}. All points on $\tilde{C}$ between a^{j+1} and a^l all differ on only the n-th coordinate. That means they can be replaced by the subpath $\langle a^{j+1}, a^l \rangle$ with equal utility change. Moreover, a^{j+1} and a^l also differ on only the n-th coordinate. So $(b^{j+1}, 1) = (b^l, 1)$. We conclude that the subpath $\langle a^{j+1}, (b^{j+1}, 1) \rangle$ of $\tilde{C}$ is followed by a subpath in the opposite direction, which has equal and opposite utility change.

If we remove all these subpaths that cancel each other out from $\tilde{C}$, we are left with a cycle $C' \subseteq \{a \in N : a_n = 1\}$, as desired. □

3.3 Examples

Global Connection Game

3.3.1 Theorem. *All instances of the Global Connection Game are potential games.*

Proof. Let GCG(G) be an instance of the Global Connection Game and N a Nash grid for GCG(G). We will prove that every elementary strategy cycle has utility change 0. Theorem 3.2.3 then implies that GCG(G) is a potential game.

Let $C = \langle a^1, a^2, a^3, a^4, a^5 \rangle$ be an elementary strategy cycle. The only non-trivial case is where $p(a^1, a^2) = p(a^3, a^4) = i$ and $p(a^2, a^3) = p(a^4, a^5) = j$, for some $i \neq j$. Consider an edge $e \in E(G)$. We have:

$$\begin{aligned}
\mathrm{Cost}_e(a^2) - \mathrm{Cost}_e(a^1) &= c_e \left(\frac{1}{n_e(a^1) + x} - \frac{1}{n_e(a^1)} \right) \\
\mathrm{Cost}_e(a^3) - \mathrm{Cost}_e(a^2) &= c_e \left(\frac{1}{n_e(a^1) + x + y} - \frac{1}{n_e(a^1) + x} \right) \\
\mathrm{Cost}_e(a^4) - \mathrm{Cost}_e(a^3) &= c_e \left(\frac{1}{n_e(a^1) + y} - \frac{1}{n_e(a^1) + x + y} \right) \\
\mathrm{Cost}_e(a^5) - \mathrm{Cost}_e(a^4) &= c_e \left(\frac{1}{n_e(a^1)} - \frac{1}{n_e(a^1) + y} \right),
\end{aligned}$$

where x, $y \in \{-1, 0, 1\}$. We set any fraction where the denominator is 0 to 0. The differences sum op to 0. It follows that

$$u(C) = \sum_{e \in E(G)} \sum_{k=1}^{4} \left(\mathrm{Cost}_e(a^{k+1}) - \mathrm{Cost}_e(a^k) \right) = 0.$$

We conclude that all elementary strategy cycles have zero utility change. □

Of course, we can also try to explicitly find a potential function for G to prove that it is a potential game:

Proof (of Theorem 3.3.1). Let GCG(G) be an instance of the Global Connection Game. For each edge $e \in E(G)$ define the function $\Psi_e : \mathcal{P} \to \mathbb{R}$ by

$$\Psi_e(s) = c_e \cdot \mathcal{H}_{n_e(s)},$$ [1]

Define the function $\Psi : \mathcal{P} \to \mathbb{R}$ by

$$\Psi(s) = \sum_{e \in E(G)} \Psi_e(s).$$

We will prove that Ψ is a potential function for GCG(G), which implies that GCG(G) is a potential game.

Let s and s' be two strategies that differ only on the i-th coordinate. Consider an edge $e \in E(H)$. We distinguish three situations:

- *e appears in both or neither s_i and s'_i.* In this case $n_e(s) = n_e(s')$, so player i pays the same for e in both strategies and $\Psi_e(s) = \Psi_e(s')$.

[1] $\mathcal{H}_n = \sum_1^n 1/n$

- *e is in s_i but not in s'_i.* If player i moves from strategy s to s', he saves $\mathrm{Cost}_e(s) = c_e/n_e(s)$ on edge e alone by doing so. This is precisely the decrease for Ψ_e:

$$\Psi_e(s) - \Psi_e(s') = c_e \cdot \mathcal{H}_{n_e(s)} - c_e \cdot \mathcal{H}_{n_e(s)-1} = c_e/n_e(s).$$

- *e is in s'_i but not in s_i.* Changing from strategy s to s' costs an additional $\mathrm{Cost}_e(s') = c_e/(n_e(s)+1)$ for player i. This is also the difference between $\Psi_e(s)$ and $\Psi_e(s')$.

In conclusion, every Ψ_e equals the change cost for player i due to e as he goes from s to s'. Since Ψ is simply the sum of the Ψ_e, and the utility for player i is simply the negative of the sum of all edge costs, this proves the theorem. □

Basic Utility Game

3.3.2 Theorem. *Let G be a Basic Utility Game. The negative of its social welfare function V is a potential function, so G is a potential game.*

Proof. Let $s \in S$ be a strategy vector for G and suppose player i changes his strategy from s_i to s'_i. Define $s' = (s^{-i}, s'_i)$.[2] We need to prove that

$$V(s') - V(s) = \alpha_i(s') - \alpha_i(s).$$

This follows directly from property (3′) of Definition 2.5.1:

$$\alpha_i(s') - \alpha_i(s) = V(s') - V(s^{-i}) - \big(V(s) - V(s^{-i})\big) = V(s') - V(s).$$

□

Atomic Routing Game In this paragraph we prove that an instance of the Atomic Routing Game is a potential game if it satisfies one of two conditions: the instance is unweighted or all edge cost functions are affine.

3.3.3 Theorem. *Let (N, r, c) be an instance of the Atomic Routing Game, where $r_i = R$ for each i and some $R \in \mathbb{R}_{>0}$. The function $\Phi : \mathbb{R}^{\#\mathcal{P}}_{\geq 0} \to \mathbb{R}_{\geq 0}$ given by*

$$\Phi(f) = \sum_{e \in E} \sum_{i=1}^{f_e} c_e(i)$$

is a potential function for (N, r, c). Consequently, (N, r, c) is a potential game.

Proof. Consider a flow f on N. Suppose player i changes its strategy from path P_i to P'_i, resulting in the new flow f'. The change in Φ is equal to

$$\Phi(f') - \Phi(f) = \sum_{e \in P'_i \setminus P_i} \sum_{i=1}^{R} c_e(f_e + i) - \sum_{e \in P_i \setminus P'_i} \sum_{i=0}^{R-1} c_e(f_e - i). \tag{3.3.1}$$

[2]See Appendix B for the definition of this notation.

On the other hand, the change in cost for player i is equal to[3]

$$c_{P'_i}(f') - c_{P_i}(f) = \sum_{e \in P'_i \setminus P_i} c_e(f_e + R) - \sum_{e \in P_i \setminus P'_i} c_e(f_e). \tag{3.3.2}$$

We want to prove that (3.3.1) and (3.3.2) are the same expressions. To see this, consider the situation where player i doesn't transfer his flow from P_i to P'_i all at once, but one unit of flow at a time. We get R flows $f^{(0)} = f, f^{(1)}, \ldots, f^{(R-1)}, f^{(R)} = f'$ where in flow $f^{(n)}$ player i routes n units of flow through path P'_i and $R - n$ units through P_i.

In this situation the total change in cost for player i can be expressed as

$$c_{P'_i}(f') - c_{P_i}(f) = \sum_{n=1}^{R} c_{P'_i}(f^{(n)}) - c_{P_i}(f^{(n-1)}). \tag{3.3.3}$$

Each element in the sum is equal to

$$c_{P'_i}(f^{(n)}) - c_{P_i}(f^{(n-1)}) = \sum_{e \in P'_i \setminus P_i} c_e(f_e + n) - \sum_{e \in P_i \setminus P'_i} c_e(f_e - (n-1)).$$

Plugging this result in the sum in expression (3.3.3) yields the expression (3.3.2). □

Again, we can also use Theorem 3.2.2 to prove that unweighted instances are potential games. In fact, we can use practically the same reasoning as for the Global Connection Game (Theorem 3.3.1). This is because the only important difference between these games, is that cost functions are non-decreasing in atomic routing and decreasing in the global connection game. This fact, however, is not relevant in the proof of Theorem 3.3.1.

With this is mind, we will prove another statement, namely that any atomic instance with affine cost functions is a potential game.

3.3.4 Theorem. *Let (N, r, c) be an instance of the Atomic Routing Game, where each c_e is affine. This instance is a potential game.*

Proof. The proof runs along the same lines as the proof of Theorem 3.3.1. Let $C = \langle a^1, a^2, a^3, a^4, a^5 \rangle$ be an elementary strategy cycle. Again, we consider only the nontrivial case where $p(a^1, a^2) = p(a^3, a^4) = i$ and $p(a^2, a^3) = p(a^4, a^5) = j$, for some $i \neq j$.

Let f be the flow from a^1. Each edge $e \in E(N)$ satisfies the following equations:

$$\begin{aligned}
r_i \cdot c_e(a^2) - r_i \cdot c_e(a^1) &= r_i \left(c_e(f_e + x) - c_e(f_e)\right) &&= r_i \cdot c_e(x) \\
r_j \cdot c_e(a^3) - r_j \cdot c_e(a^2) &= r_j \left(c_e(f_e + x + y) - c_e(f_e + x)\right) &&= r_j \cdot c_e(y) \\
r_i \cdot c_e(a^4) - r_i \cdot c_e(a^3) &= r_i \left(c_e(f_e + y) - c_e(f_e + x + y)\right) &&= -r_i \cdot c_e(x) \\
r_j \cdot c_e(a^5) - r_j \cdot c_e(a^4) &= r_j \left(c_e(f_e) - c_e(f_e + y)\right) &&= -r_j \cdot c_e(y),
\end{aligned}$$

where $x \in \{-r_i, 0, r_i\}$ and $y \in \{-r_j, 0, r_j\}$. These equation hold because c_e is affine. The rest of the proof follows the same reasoning as the proof of Theorem 3.3.1. □

[3]Since the instance is unweighted, we don't have to multiply with player i's traffic rate here.

3.4 Properties of potential games

A strong property of potential games is that they always have a Nash equilibrium. This follows quite easily from the definition.

3.4.1 Theorem. *If G is a potential game with potential function Φ, then any strategy that minimizes Φ is a Nash equilibrium.*

Proof. Let s be a strategy vector that minimizes Φ. Consider the vector $s' = (s^{-i}, s'_i) \in S$, where $s'_i \in S_i$. The fact that s minimizes Φ implies the inequality $\Phi(s') \geq \Phi(s)$. From (3.2.1) it follows that $u_i(s') \leq u_i(s)$, i.e. player i's deviation from s_i to s'_i is not profitable. Therefore, s is a Nash equilibrium. □

3.4.2 Corollary. *Every potential game has a Nash equilibrium.*

Proof. A potential game is finite, so there are only a finite number of strategies. Therefore the potential function of a potential game has a minimum. From Theorem 3.4.1 it follows that a Nash equilibrium exists. □

A slight change in the proof of Theorem 3.4.1 gives a nice result for convergence to Nash equilibria. When players use *best response dynamics*, i.e. at each step one player changes to a strategy that gives him the highest utility currently possible, then this process will always converge to a Nash equilibrium.

3.4.3 Theorem. *In every potential game, best response dynamics always converge to a Nash equilibrium.*

Proof. If a player changes to a strategy with higher utility, the game's potential function decreases. Since there are only a finite number of strategies, at some point the players arrive at a strategy where no unilateral deviation can decrease the value of the potential function. From (3.2.1) it follows that no player can find a strategy that increases his utility. The current strategy, then, is a Nash equilibrium. □

The Price of Stability of a potential game can be easily bounded from above if the potential function and social cost function behave 'about the same':

3.4.4 Theorem. *Let $G = (A, S, U)$ be a potential game with potential function Φ. Suppose for each strategy vector $s \in S$ the following inequalities hold for some constants A, $B > 0$:*

$$\frac{\text{Cost}(s)}{A} \leq \Phi(s) \leq B \cdot \text{Cost}(s).$$

Then the Price of Stability of G is at most AB.

Proof. Consider an $s \in S$ that minimizes Φ. Let $s_{\text{opt}} \in S$ be an optimal strategy vector. Due to Theorem 3.4.1, s is a Nash equilibrium. Moreover, the following inequalities hold:

$$\frac{\text{Cost}(s)}{A} \leq \Phi(s) \leq \Phi(s_{\text{opt}}) \leq B \cdot \text{Cost}(s_{\text{opt}}).$$

The first and last inequalites are true by the condition stated in this theorem, while the middle inequality is true because s minimizes Φ. Rearranging yields

$$\frac{\text{Cost}(s)}{\text{Cost}(s_{\text{opt}})} \leq AB.$$

Since s is a Nash equilibrium, it follows that $\sigma(G) \leq AB$. □

3.5 Applications to specific games

The Global Connection Game satisfies the condition in Theorem 3.4.4, yielding the bound on the Price of Stability as mentioned in Section 2.3.3.

3.5.1 Lemma. *Let G be an instance of the Global Connection Game and let Ψ be its corresponding potential function, as defined in Theorem 3.3.1. The following inequalities hold:*

$$\text{Cost}(S) \leq \Psi(S) \leq \mathcal{H}_n \text{Cost}(S).$$

Proof. Let S be a strategy vector for G. For any edge e that a path in S uses, the inequalities $1 \leq n_e \leq n$ hold, and consequently the following inequalities hold as well:

$$c_e \leq \Psi_e(S) = c_e \cdot \mathcal{H}_{n_e} \leq c_e \cdot \mathcal{H}_n.$$

If e is not used in S, both its contribution to the social cost and Ψ_e are zero. Since the social cost is the sum of the c_e used in S and Ψ is the sum of the Ψ_e, the functions as mentioned in this lemma hold. □

3.5.2 Theorem. *The Price of Stability of any instance of the Global Connection Game is at most $\mathcal{H}_n$. This is a strict upper bound.*

Proof. The strictness follows from Example 2.3.2. The upper bound itself follows immediatly from Theorem 3.4.4 and Lemma 3.5.1. □

3.5.3 Theorem. *Every Basic Utility Game G satisfies $\sigma(G) = 1$. Furthermore, best response dynamics always converge to a Nash equilibrium.*

Proof. Since the social welfare function of G is the negative of a potential function for G, any optimal strategy s_{opt} for G is a Nash equilibrium (see Theorems 3.3.2 and 3.4.1). The statement about best response dynamics follows from Theorem 3.4.3. □

Appendix A

Biconnected components

This appendix relates to Section 2.2.4 on the Price of Anarchy of the Local Connection Game. Here we prove several technical lemmas which are used to prove the bound on the Local Connection Game's Price of Anarchy found at the end of section 2.2.4.

The central fact used to prove the lower bound in Lemma 2.2.12 is that nodes in a biconnected component H (maximal subgraph without a cut vertex[1]) in equilibrium networks are never far from nodes with degree 3 in H. More precisely, if $\alpha > 19n$, then nodes in H are always at most 11 edges removed from nodes with degree 3 in H (Lemma A.4). The proof of this result follows three steps.

Firstly, we will prove that the cycles in H are at least 22 edges long when $\alpha > 19n$ (Lemma A.1).

Secondly, we look at a cycle C in H and consider two nodes u and v on C which are 11 edges apart from each other. For contradiction we assume all nodes on C between u and v have degree 2 in H. We prove that this means that, if $x_0, \ldots, x_9$ are the nodes in C between u and v and $x_{10} := v$, the edge $\{x_i, x_{i+1}\}$ is bought by x_i for each $i = 1, \ldots, 9$ (Lemma A.2).

Finally, we prove that such a path can have length at most 8 (Lemma A.3), leading to a contradiction. It follows that some node on C between u and v has degree 3 in H.

A.1 Lemma. *Let C be a cycle in $N(s_{\text{Nash}})$, where s_{Nash} is a Nash equilibrium for* $\text{LCG}(n, \alpha)$. *Then* $\text{length}(C) \geq \frac{\alpha}{n-2} + 2$.

Proof. Consider an edge $\{u, v\} \in C$ built by u. If u decides not to build $\{u, v\}$, his usage costs increase by at most $(n-2) \cdot (\text{length}(C) - 2)$. This is because at worst the distance to all nodes, except u and his neighbor in C besides v, will increase to the length of the detour to v via C.

Player u cannot profit by removing $\{u, v\}$. This means the inequality $(n-2) \cdot (\text{length}(C) - 2) \geq \alpha$ must be satisfied. Rearranging yields $\text{length}(C) \geq \frac{\alpha}{n-2} + 2$. ☐

For the next two lemmas, we introduce the following notation. For a node x in a biconnected component H of a network $N = (V, E)$, let $D_H(x)$ be the

[1]a cut vertex is a vertex whose removal makes the graph disconnected

set of nodes $v \in V$ for which a path P to x exists such that x is the only node in H that is contained in P. This definition includes the single-node path, so $x \in D_H(x)$. Note that $D_H(x) \cap D_H(y) = \emptyset$ for each x,$y \in V(H)$, since H is maximal and therefore contains the nodes in all paths between x and y.

A.2 Lemma. *Let s_{Nash} be a Nash equilibrium for $\mathrm{LCG}(n, \alpha)$, H a biconnected component in $N(s_{\mathrm{Nash}})$ and u,$v \in V(H)$ be such that $\mathrm{dist}(u, v) \geq 3$. Suppose there is a shortest path P between u and v such that u buys $\{u, x\} \in P$ and v buys $\{y, v\} \in P$. Then $\deg_H(x) \geq 3$ or $\deg_H(y) \geq 3$.*

Proof. Suppose $\deg_H(x) = \deg_H(y) = 2$. Assume $|D_H(x)| \leq |D_H(y)|$. Suppose u changes his strategy by removing $\{u, x\}$ and building $\{u, z\}$, where z is the other neighbor of x in H. By building $\{u, z\}$, the distance from u to v and all nodes in $D_H(y)$ is decreased by 1, yielding $|D_H(y)| + 1$ in savings for u.

Removing $\{u, x\}$, on the other hand, increases the distance to all nodes in $D_H(x)$ by 1, increasing u's usage costs by $|D_H(x)|$. No other distances are increased. Since $|D_H(x)| < |D_H(y)| + 1$, u profits, so s_{Nash} is not a Nash equilibrium. □

A.3 Lemma. *Let H be a biconnected component in an equilibrium network N of $\mathrm{LCG}(n, \alpha)$. Let $\langle x_0, \ldots, x_k \rangle$ be a path in H that is not a cycle such that $\deg_H(x_i) = 2$ for all $i \in \{0, \ldots, k\}$ and x_i buys $\{x_i, x_{i+1}\}$ for all $i \in \{0, \ldots, k-1\}$. Then $k \leq 8$.*

Proof. Consider a path $\langle x_0, \ldots, x_k \rangle$ in H with properties as stated above of maximal size, but assume for contradiction $k \geq 9$. Let $x_{k+1} \in V(H)$ be the neighbor of x_k in H besides x_{k-1}. Let T be a BFS-tree of N rooted in x_{k-1} and let A be the set of nodes containing x_{k+1} and his descendants in T.

Firstly, we prove the following lower bound:

$$|D_H(x_1)| \geq 2^{\lceil k/2 \rceil - 2} |A|. \tag{A.0.1}$$

If x_{k-1} buys edge $\{x_{k-1}, x_{k+1}\}$ instead of $\{x_{k-1}, x_k\}$, his usage costs to all nodes in $D_H(x_k)$ will increase by 1, while his usage costs to all other nodes will not increase. Moreover, his usage costs to the nodes in A will decrease by 1. This is because, as T is a BFS-tree of N rooted in x_{k-1}, $\mathrm{dist}(x_{k-1}, u) = \mathrm{dist}(x_{k-1}, x_{k+1}) + \mathrm{dist}(x_{k+1}, u)$ for all $u \in A$. Since N is an equilibrium network, we must have $|D_H(x_k)| \geq |A|$.

For $i \in \{1, \ldots, k-1\}$, if x_{i-1} buys edge $\{x_{i-1}, x_{i+1}\}$ instead of $\{x_{i-1}, x_i\}$, then x_{i-1}'s usage costs to the nodes in $D_H(x_i)$ will increase by 1, while usage costs to other nodes will not increase. Usage costs to a node x_j will definitely decrease if there can be no alternative path from x_{i-1} to x_j of length at most $j - i$. Note that the smallest possible alternative path exists when $x_0 = x_{k+1}$. The length of this path is $k + i - j$. So for $j \leq \lceil k/2 \rceil + i - 1$ the node x_{i-1} will decrease his usage costs by $|D_H(x_j)|$. Again, since N is an equilibrium network, we must have

$$|D_H(x_i)| \geq \left| \bigcup_{j=i+1}^{\min\{\lceil k/2 \rceil + i - 1, k\}} D_H(x_j) \right|$$

$$= \sum_{j=i+1}^{\min\{\lceil k/2 \rceil + i - 1, k\}} |D_H(x_j)|, \tag{A.0.2}$$

where the last equality holds because the D_H are pairwise disjoint. If we take $i = \lfloor k/2 \rfloor + 1$ we get, by induction,

$$|D_H(x_1)| \geq |D_H(x_{\lfloor k/2 \rfloor+1})| \geq 2^{\lceil k/2 \rceil - 2}|A|.$$

Secondly, we prove that $\deg_H(x_{k+1}) \geq 3$. Since $\langle x_0, \ldots, x_k \rangle$ is of maximal size, precisely one of the following three cases holds. Either $\deg_H(x_{k+1}) \geq 3$ or x_{k+1} buys $\{x_k, x_{k+1}\}$ or $x_{k+1} = x_0$ and x_k buys $\{x_k, x_{k+1}\}$. In the first case we are done. In the second case Lemma A.2 implies $\deg_H(x_{k-1}) \geq 3$ or $\deg_H(x_k) \geq 3$, which is a contradiction. In the third case, consider what happens when x_k buys edge $\{x_k, x_1\}$ instead of $\{x_k, x_0\}$. Then his usage costs to the nodes in $D_H(x_0)$ will increase by 1, his usage costs to the nodes in $D_H(x_1)$ will decrease by 1 and the distances to all other nodes will not increase. Observe that $D_H(x_0) \subseteq A$, so $|A| \geq |D_H(x_0)|$. As N is an equilibrium network, it follows that $|D_H(x_0)| \geq |D_H(x_1)|$. However, from (A.0.1) it follows that $|D_H(x_1)| > |A|$, so this case also leads to a contradiction.

Thirdly, we prove that x_{k+1} buys the edges to all his children in T. Suppose there is a node $u \in V(N)$ that is a child of x_{k+1} in T and that buys $\{x_{k+1}, u\}$. Consider the situation where u buys $\{x_{k-5}, u\}$ instead of $\{x_{k+1}, u\}$. (Note that there is no edge $\{x_{k-5}, u\}$ in N since $\deg_H(x_{k-5}) = 2$ and $u \neq x_{k-6}$,x_{k-4} because those nodes also have H-degree 2.)

The usage costs for u to all nodes in $D_H(x_{k-5})$, $D_H(x_{k-4})$ and $D_H(x_{k-3})$ will decrease by at least 2, while the usage costs to all nodes in $D_H(x_{k-1})$ and $D_H(x_k)$ and possibly some nodes in A will increase by at most 6. The distance from u to any other node v will not be increased. This is because if $\text{dist}(x_{k+1}, v) < \text{dist}(x_{k-5}, v)$, then the v will be a descendent of x_{k+1} in T, so $v \in A$.

Because N is an equilibrium network, it follows that

$$2|D_H(x_{k-5}) \cup D_H(x_{k-4}) \cup D_H(x_{k-3})| \leq 6|D_H(x_{k-1}) \cup D_H(x_k) \cup A|.$$

From (A.0.2) we deduce that

$$2|D_H(x_{k-5}) \cup D_H(x_{k-4}) \cup D_H(x_{k-3})| > 18|D_H(x_{k-1})|,$$

while also from (A.0.2) we deduce that $|A| \leq |D_H(x_k)| \leq |D_H(x_{k-1})|$, so

$$6|D_H(x_{k-1}) \cup D_H(x_k) \cup A| \leq 18|D_H(x_{k-1})|.$$

Putting these inequalities together yields a contradiction.

Finally, we derive a contradiction by changing x_{k+1}'s strategy. Suppose x_{k+1} buys $\{x_1, x_{k+1}\}$ and removes the edges to all his children in H. Let $v \in V(H)$ be a node for which $\text{dist}(x_{k+1}, v)$ increases by this change in strategy. Then we must have $\text{dist}(u, v) = \text{dist}(x_1, v) - 1$ for some child u of x_{k+1} in T. This means v is a descendant of x_{k+1} in T, i.e. $v \in A$. So the distance to any vertex in $V(N) \setminus A$ will not be increased.

Since x_{k+1} is not a cut vertex, it can still reach each vertex in A. The distance to any such vertex will increase by at most $2 \cdot \text{diam}(H)$. By deleting edges, x_{k+1} will save at least α. It follows that we must have $\alpha \leq 2 \cdot \text{diam}(H) \cdot |A|$.

Next, we prove that $(\text{rad}(H) - 1)|D_H(x_1)| \leq \alpha$.[2] Suppose an edge with distance $\text{rad}(H)$ from x_1 buys an edge to x_1. This will cost α, but decrease his

[2] $\text{rad}(G) = \min_{u \in V} \max_{v \in V} \text{dist}(u, v)$. Here $G = (V, E)$ is a connected graph.

usage costs by at least $(\mathrm{rad}(H)-1)|D_H(x_1)|$. The equilibrium constraint yields the desired inequality.

By applying inequality (A.0.1), we get

$$\begin{aligned}(\mathrm{rad}(H)-1)2^{\lceil k/2\rceil-2}|A| &\leq (\mathrm{rad}(H)-1)|D_H(x_1)| \\ &\leq 2\cdot \mathrm{diam}(H)\cdot |A| \\ &\leq 4\cdot \mathrm{rad}(H)\cdot |A|.\end{aligned}$$

Consequently,

$$\mathrm{rad}(H) \leq \frac{2^{\lceil k/2\rceil-4}}{2^{\lceil k/2\rceil-4}-1} \leq 2.$$

However, since $k \geq 9$ we have $\mathrm{rad}(H) > 2$; contradiction. □

A.4 Lemma. *Let s_{Nash} be a Nash equilibrium for* $\mathrm{LCG}(n,\alpha)$*, where $\alpha > 19n$. Let H be a biconnected component in $N(s_{\mathrm{Nash}})$. For every $u \in V(H)$ there is a $v \in V(H)$ with* $\mathrm{dist}(u,v) \leq 11$ *and* $\deg_H(v) \geq 3$.

Proof. Let $\{u,u'\}$ be an edge in H and $C \subseteq H$ a cycle for which $\{u,u'\} \in C$. Assume u buys the edge $\{u,u'\}$. We will prove that there is a $v \in V(C)$ with $\mathrm{dist}(u,v) = \mathrm{dist}(u',v)+1 \leq 11$ and $\deg_H(v) \geq 3$, thereby proving the theorem.

Note that, by Lemma A.1, $\mathrm{length}(C) \geq 22$. It follows that for any $v \in V(C)$ we have $\mathrm{dist}_C(u,v) \leq 11$ and $\mathrm{dist}_C(u,v) = \mathrm{dist}(u,v)$. This is because otherwise H would contain a cycle C' with $\mathrm{length}(C') = \mathrm{dist}_C(u,v) + \mathrm{dist}(u,v) < 22$, in contradiction with Lemma A.1.

Given $v \in V(C)$, we denote P_{uv} for a shortest path between u and v contained in C. Suppose there is a node $v \in V(C)$ such that $\mathrm{dist}(u,v) \geq 3$, $\{u,u'\} \in P_{uv}$ and v buys the edge $\{y,v\} \in P_{uv}$. From Lemma A.2 it follows that there is a $w \in P_{uv}$ such that $\deg_H(w) \geq 3$.

Suppose, on the other hand, that for any node $v \in V(C)$ with $\mathrm{dist}(u,v) \geq 3$ and $\{u,u'\} \in P_{uv}$, the edge $\{y,v\} \in P_{uv}$ is not bought by v. In that case, there is a path $P \subseteq C$ of length 9 running along nodes $\{x_0,\ldots,x_9\}$ such that $\{u',x_0\} \in C$ and each edge $\{x_i,x_{i+1}\}$ is paid for by x_i. By Lemma A.3, this is not possible. □

Appendix B

Finite strategic games

A **finite strategic game** G is a triple $G = (A, S = \Pi_{a \in A} S_a, U)$, where A is a finite set of **players**, S_a is a finite set called the **strategy set** for player a and U is a set of **utility functions** $u_a : S \to \mathbb{R}$, one for each player $a \in A$. Given a utility function u_a, the **cost function** $\text{Cost}_a : S \to \mathbb{R}$ of a player $a \in A$ is defined by $\text{Cost}_a \equiv -u_a$.

A vector $s \in S$ is called a **strategy vector**. For convenience we sometimes denote a strategy vector $s \in S$ by $s = (s^{-a}, s_a)$, where the vector $s^{-a} \in \Pi_{b \in A \setminus \{a\}} S_b$ satisfies $s_b^{-a} = s_b$ for each $b \in A \setminus \{a\}$. A strategy vector s_{Nash} is called a **Nash equilibrium** if

$$\forall a \in A \forall s'_a \in S_a : u_a(s_{\text{Nash}}) \geq u_a(s^{-a}_{\text{Nash}}, s'_a),$$

i.e. no player can improve his utility by unilaterally changing his strategy.

To quantify the inefficiency of the Nash equilibria of G, we need to compare the 'social' cost or utility of the Nash equilibria to an 'optimal' value. For this, we define a **social cost function** $\text{Cost}_G : S \to \mathbb{R}$ or, equivalently, a **social utility function** $u_G : S \to \mathbb{R}$. The social cost function is most often defined as $\text{Cost}_G \equiv \sum_{a \in A} \text{Cost}_a$. Usually we omit the subscript G, since it is clear from the context which game we're considering. A vector $s_{\text{opt}} \in S$ is called an **optimal strategy** if

$$\text{Cost}(s_{\text{opt}}) = \min_{s \in S} \text{Cost}(s).$$

The **Price of Anarchy** $\rho(G)$ and the **Price of Stability** $\sigma(G)$ of G are defined as

$$\rho(G) = \max_{s_{\text{Nash}} \in S} \frac{\text{Cost}(s_{\text{Nash}})}{\text{Cost}(s_{\text{opt}})},$$
$$\sigma(G) = \min_{s_{\text{Nash}} \in S} \frac{\text{Cost}(s_{\text{Nash}})}{\text{Cost}(s_{\text{opt}})},$$

where each s_{Nash} is a Nash equilibrium, $s_{\text{opt}} \in S$ is an optimal strategy, $0/0$ is set to 1 and $c/0$ where $c > 0$ to $+\infty$.

Index

Bibliography

[1] Nir Andelman, Michal Feldman, and Yishay Mansour, *Strong price of anarchy*, Proceedings of the eighteenth annual acm-siam symposium on discrete algorithms, 2007, pp. 189–198.

[2] Elliot Anshelevich, Anirban Dasgupta, Jon Kleinberg, Éva Tardos, Tom Wexler, and Tim Roughgarden, *The price of stability for network design with fair cost allocation*, SIAM Journal on Computing **38** (2008), no. 4, 1602–1623.

[3] Baruch Awerbuch, Yossi Azar, and Amir Epstein, *The price of routing unsplittable flow*, Proceedings of the thirty-seventh annual acm symposium on theory of computing, 2005, pp. 57–66.

[4] Alex Fabrikant, Ankur Luthra, Elitza Maneva, Christos H. Papadimitriou, and Scott Shenker, *On a network creation game*, Proceedings of the twenty-second annual symposium on principles of distributed computing, 2003, pp. 347–351.

[5] Michel Goemans, Adrian Vetta, and Vahab Mirrokni, *Sink equilibria and convergence*, In focs, 2005.

[6] Matúš Mihalák and Jan Christoph Schlegel, *The price of anarchy in network creation games is (mostly) constant*, Proceedings of the third international conference on algorithmic game theory, 2010, pp. 276–287.

[7] Dov Monderer and Lloyd S. Shapley, *Potential games*, Games and Economic Behavior **14** (1996), no. 1, 124 –143.

[8] Herv Moulin, Scott Shenker, Jel D, Stimulating Conversations Raj Deb, Bhaskar Dutta, Joan Feigenbaum, and Wolfgang Pesendorfer, *Strategyproof sharing of submodular costs: budget balance versus efficiency* **18** (1999), 511–533.

[9] John F. Nash, *Equilibrium points in n-person games*, Proceedings of the National Academy of Sciences **36** (1950), no. 1, 48–49, available at `http://www.pnas.org/content/36/1/48.full.pdf+html`.

[10] Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani (eds.) *Algorithmic game theory*, Cambridge University Press, New York, 2007.

[11] Christos Papadimitriou, *Algorithmic game theory: A snapshot*, Automata, languages and programming, 2009, pp. 3–11.

[12] Pablo Pedregal, *Introduction to optimization*, Texts in Applied Mathematics, Springer-Verlag, New York, 2004.

[13] Arthur C. Pigou, *The economics of welfare*, Macmillan and Co. London, 1932.

[14] Tim Roughgarden, *Selfish routing*, Ph.D. Thesis, 2002.

[15] ———, *Algorithmic game theory*, Commun. ACM **53** (July 2010), 78–86.

[16] Tim Roughgarden and Éva Tardos, *How bad is selfish routing?* J. ACM **49** (2002March), 236–259.

[17] Neil Spring, Ratul Mahajan, and Thomas Anderson, *The causes of path inflation*, Proceedings of the 2003 conference on applications, technologies, architectures, and protocols for computer communications, 2003, pp. 113–124.

[18] Adrian Vetta, *Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions*, Proceedings of the 43rd symposium on foundations of computer science, 2002, pp. 416–.

[19] J. Wardrop, *Some theoretical aspects of road traffic research*, Proceedings of the Institution of Civil Engineers, Part II **1** (1952), no. 36, 352–362.