Networks and Distributed Operation

The Price of Anarchy

in Non-Atomic Routing and Network Formation

Dissertation

zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften
(Dr. rer. nat.)

der Technischen Fakultät
der Christian-Albrechts-Universität zu Kiel

Dipl.-Math. Lasse Kliemann

Kiel

2009

1. Gutachter: Prof. Dr. Anand Srivastav
Christian-Albrechts-Universität Kiel

2. Gutachter: Prof. Dr. Klaus Jansen
Christian-Albrechts-Universität Kiel

3. Gutachter: Prof. Dr. Matthias Müller-Hannemann
Martin-Luther-Universität Halle-Wittenberg

Datum der mündlichen Prüfung: 11. Januar 2010

Networks and Distributed Operation

The Price of Anarchy

in Non-Atomic Routing and Network Formation

Lasse Kliemann

Errata and revised versions can be found via

http://lasse-kliemann.name

or by using the permanent ID.

Permanent ID of this document:

d811d50d-931a-4413-bffe-1cd6a2e1d84c

This version is dated 2010-10-20.

To search for (newer versions of) this document, use the search string:

```
Permanent ID of this document: d811d50d-931a-4413-bffe-1cd6a2e1d84c
```

When citing this document, please use the document ID in the following form: "Document ID: d811d50d-931a-4413-bffe-1cd6a2e1d84c". You can also use the date to refer to a particular version of this document.

Please do *not* use the phrase "Permanent ID of this document" when citing, since that would make it impractical to find this document using the search string given above.

Contents

Acknowledgments

I thank my doctoral adviser Prof. Dr. Srivastav for introducing me to the subject, supporting me throughout my studies, and for providing me with many opportunities for research and contacts.

I thank Ole Kliemann for his invaluable work on the software implementation for computing Nash equilibria and the price of anarchy, which he carried out as a part of the DFG Priority Program 1126 "Algorithmics of Large and Complex Networks". I also thank him for helpful discussions on Chapter 2.

I thank Andreas Baltz and Sandro Esquivel for joint work on selfish multicast routing in 2005 and 2006.

I thank Michael Gnewuch for helpful discussions on Chapter 1.

I thank Gerold Jäger for helpful discussions on Chapter 5.

I thank the Rechenzentrum Universität Kiel for providing many weeks of computing time.

I thank the Dr. Robert-Gedächtnisstiftung and the Christian-Albrechts-Universität zu Kiel for providing me with a scholarship.

I thank the Deutsche Forschungsgemeinschaft for financial support through Priority Program 1126 "Algorithmics of Large and Complex Networks" (Grant SR7-3) and Priority Program 1307 "Algorithm Engineering" (Grant SR7/12-1).

And I thank everyone else who supported me during my studies, research, and writing of this thesis.

Introduction

Networks and Distributed Operation

What is the worst overall degradation due to the absence of central authority or a lack of cooperation when individuals use or form a network? "Distributed operation" is our positive notion of "absence of central authority" and "lack of cooperation". The "price of anarchy" is the measure for the worst overall degradation due to distributed operation. All scenarios studied in this thesis have in common that individuals have to allocate resources in order to perform certain tasks, and that each possible configuration can be evaluated from the perspective of each individual and from a global perspective. We use the term *players* to refer to the individuals. Evaluation from player-perspective assigns to each player a single non-negative real number, called the *individual cost* of that player. Evaluation from a global perspective is expressed by a single non-negative number, called *social cost*. Such a setting is referred to as a "game" and the efforts to its comprehension as "game theory", although there is no relation to the recreational activities.[1]

In our models, there is always an optimal configuration, i.e., with minimal social cost, denoted OPT. We compare OPT to the social cost of equilibria, i.e., configurations in which players do not have a way to improve their individual cost, or at least they cannot see such a way. How far players can "see" and what influence they have is different for different models. In other words, optimal configurations are optima in

[1] Quoting Myerson [67]: "'Conflict Analysis' or 'interactive decision theory' might be more descriptively accurate names for the subject, but the name 'game theory' seems to be here to stay."

the usual sense: we have an objective function, the social cost, and if a central authority was in charge, it would try to minimize it. Whereas, in a distributed model, players each optimize from their own point of view using the individual cost as objective function. This has lead to "system-optimized" and "user-optimized" attributes, which are synonymous to our terms "optimal" and "in equilibrium".

We restrict to models that are non-cooperative or only minimally cooperative. That is, players do not form coalitions (or if they do, then only small ones). We also describe such situations as having a high degree of distribution – each player acts mostly on her own. Generally, both the absence of central authority and a lack of cooperation can be the reason for a tension between optimum and equilibrium. For instance, in a network routing scenario, a central authority might sacrifice some of the players and force them to take a slow route – with the effect that the rest of the players experience less congestion and enjoy very fast routes. Such a distribution can be necessary to attain an optimum. It may, in fact, happen that some players are worse off in any optimum than any player would be in an equilibrium. But also lack of cooperation alone can be the source of tension between optimality and equilibrium. We elaborate on this in Sec. 1.8.

The *price of anarchy* is the social cost of a worst-case equilibrium divided by OPT.[2] When we view distributed operation as an approximation algorithm, the price of anarchy is its approximation ratio. We study the price of anarchy with a special focus on network-related issues. All our analysis are static: we do not consider how configurations might evolve over time when players repeatedly change their strategies in order to improve their individual cost. Instead, we describe properties of equilibria and worst-case equilibria, with a special focus on their relation to optima.

In Part I, a network is given and players use the network for routing. This basic setting is also known as "selfish routing", emphasizing that players only care for their individual costs and not for the social cost. "Non-atomic routing" means that a single player has only a negligible

[2] In our models, existence of best- and worst-case equilibria is guaranteed. Moreover, we assume $\mathsf{OPT} > 0$; for our models, this is automatically the case or is enforced by small restrictions.

effect on the network, like it is the case in road traffic or when there is a large number of small transmissions in a communication network. We study multicast routing and this puts network structure more into focus than unicast routing. We show that the price of anarchy depends strongly on the network structure and prove almost tight lower and upper bounds. This part also contains a comprehensive experimental study of the price of anarchy.

In Part II, the network itself is formed in a distributed manner. Each vertex represents a player. The players can build edges to other players. Building edges incurs a cost but also has a positive effect since it improves connectivity[3] of the network. Both aspects are incorporated into the players' cost functions: the building cost and the indirect cost, related to properties of the network. This area of research is a widely open field. There seems to be an endless variety of interesting measures to be used for the indirect part of the cost functions. Moreover, there are at least two different ways in which links can be formed: in the *unilateral case*, each player may connect to any other player and pays an amount of α for each link ($\alpha > 0$ being a parameter); whereas in the *bilateral case*, the consent of both endpoints is needed, and then each of them pays α. We make a contribution by considering *robustness* aspects in a new way: we evaluate how good the network can withstand the deletion of one edge, chosen by an adversary – a classical scenario, e.g., in the area of fault-tolerant networks [3, 21, 22, 29, 48, 81]. We consider the unilateral and the bilateral case.

Aside from the price of anarchy, two further aspects are of interest; first, one asks about the structure of (worst-case) equilibria: how traffic is distributed in the routing model or how equilibrium networks look like in the network formation model. Second, when different equilibria can have different social cost, how expensive, compared to OPT, is the cheapest equilibrium? We use the *price of stability* as the measure, i.e., the social cost of an optimal equilibrium divided by OPT. We will consider structural aspects and the price of stability occasionally along the way, while our main theme remains the price of anarchy.

[3] "Connectivity" is used in a general sense here, not limited to k-connectedness of graphs. Moreover, the model introduced in Ch. 5 offers clearance for notions of connectivity where additional edges can have a negative effect.

Both parts of this thesis, Part I and Part II, are written in a way so that they can be understood independently of each other. Each of them contains an outlook on possible future research at the end. Basic terminology is collected in App. A.

Results of the Thesis

Part I. The main achievements in the first part are the introduction of a new model, namely the model of non-atomic consumption-relevance congestion games (NCRCG), almost tight lower and upper bounds on the price of anarchy in NCRCGs, and a thorough experimental study. NCRCGs are an extension of the well-known non-atomic congestion games (NCG). They are motivated by selfish multicast routing, but also interesting in their own right. They exhibit new phenomena not present in NCGs (or selfish unicast routing).

In NCGs as well as NCRCGs, players have to allocate elementary resources, called *elements*, in order to perform tasks. For example, elements can be links in a communication network and tasks can be data transmissions. Elements can only be allocated in certain combinations, called *strategies*, e.g., forming paths or trees in a network. Allocation of elements induces *congestion* for each of the elements. Each element completes its part of the task with a latency, expressed by its *element latency function*, a real function, applied to the congestion of the respective element. In other words, the element latency function of an element expresses how this element's performance reacts to congestion. Element latencies reflect back to strategies, resulting in *strategy latencies*. These are experienced by the players as the delay with which their task is completed. In a Nash equilibrium, no player has an incentive to switch the currently chosen strategy, given the decisions of other players. The NCRCG model offers a larger variety than the NCG model to express how occupation of strategies inflicts congestion and hence element latency, and how element latency inflicts strategy latency. This is the source for new phenomena and challenges.

An instance of the NCRCG model (shortly: "an NCRCG") can admit multiple Nash equilibria of different social cost, even when restricted to the special case of multicast routing with strictly increasing

link latency functions. An NCG instance, in contrast, only admits Nash equilibria of the same social cost; for strictly increasing element latency functions, all Nash equilibria of an NCG even induce the same congestions.

The price of anarchy of an NCG can be bounded with the *anarchy value* $\alpha(\mathcal{L})$ or the $\beta(\mathcal{L})$ *parameter*, both dependent only on the class $\mathcal{L}$ of occurring element latency functions. The price of anarchy of an NCRCG, however, cannot be bounded with the anarchy value or the β parameter alone, but we have to incorporate structural parameters in the bound. We prove an upper bound on the price of anarchy for s-super-homogeneous element latency functions. An element latency function ℓ is s-super-homogeneous with s being a function $s : (0,1] \longrightarrow (0,\infty)$ with $s(1) = 1$, if $\ell(\varepsilon x) \geq s(\varepsilon)\ell(x)$ for all $\varepsilon \in (0,1]$ and $x \in \mathbb{R}_{\geq 0}$. Denote function $\bar{s}(t) : [1,\infty) \longrightarrow (0,\infty), t \mapsto s(t^{-1})^{-1}$. Then s-super-homogeneity is equivalent to $\bar{s}$-sub-homogeneity, i.e., the property $\ell(tx) \leq \bar{s}(t)\ell(x)$ for all $t \in [1,\infty)$ and $x \in \mathbb{R}_{\geq 0}$. For example, polynomials of degree at most p and non-negative coefficients are $(\varepsilon \mapsto \varepsilon^p)$-super-homogeneous and also $(t \mapsto t^p)$-sub-homogeneous. Our bound on the price of anarchy is

$$\begin{cases} \frac{1}{1-\beta}\gamma\bar{s}(\gamma) & \text{for all } \gamma \\ \min\left\{\frac{1}{1-\gamma\beta}\gamma,\ \frac{1}{1-\beta}\gamma\bar{s}(\gamma)\right\} & \text{for } \gamma < 1/\beta \end{cases}.$$

Here, $\beta = \beta(\mathcal{L})$ is the known β parameter, and γ is our *new structural parameter*. It is $\gamma = 1$ for NCGs, in which case this bound collapses to the well-known $\frac{1}{1-\beta}$ bound [33, 80]. If element latency functions are polynomials of degree at most p and non-negative coefficients, denoted $\mathrm{Poly}^+(p)$, our bound is simplified and improved to

$$\begin{cases} \gamma^{p+1} & \text{if } \gamma \geq (1+p)^{\frac{1}{p}} \\ \frac{1}{1-\gamma\beta}\gamma & \text{if } \gamma \leq (1+p)^{\frac{1}{p}} \end{cases},$$

where $\beta = \beta(\mathrm{Poly}^+(p)) = p\,(p+1)^{-1-\frac{1}{p}}$.

We prove that the latter is tight up to a factor of γ. We also prove a bicriteria bound: when scaling up demands by $(1+\beta(\mathcal{L}))\,\gamma$, even

an optimum is no less costly than a Nash equilibrium for the original demands. This provides a natural extension to previous results on NCGs, where a scaling factor of $1 + \beta(\mathcal{L})$ occurs. We also show that for polynomials, the new scaling factor is the best possible up to a factor of $(1 + \beta(\mathcal{L}))\,\gamma^{\frac{1}{1+p}}$.

We lay out a theoretical foundation for computation of Nash equilibria by characterizing them as optima of a non-linear program (NLP) and devising a binary search scheme to find extreme equilibria. For NCGs, one of the central tools – not limited to computation – is the potential function. It allows characterization of Nash equilibria as optima of a convex optimization problem. Computation of extreme Nash equilibria does not pose an extra challenge in NCGs, since all Nash equilibria have the same social cost there. Surprisingly, there seems to be no way to use the potential function for NCRCGs. However, we can still give an NLP formulation based on a different technique, but unfortunately not necessarily yielding *convex* programs. To find worst-case equilibria, and hence to determine the price of anarchy, we devise a binary search scheme: in each step an NLP that carries an additional linear constraint has to be solved. The additional constraint pushes the social cost above some threshold. This threshold is adapted in each step. We have no guarantee (yet) to find a worst-case Nash equilibrium, or even to find any Nash equilibrium in this way, since it is unclear whether the involved NLPs can be solved optimally. It is to be regarded as a heuristic approach. However, our experimental results are promising.

Our experimental studies utilize this NLP formulation and binary search scheme. In total, we treat more than 10 million (small) randomly generated NCRCGs with element latency functions from $\mathrm{Poly}^{+}(p)$, for $p \in \{1, 2, 3\}$, $m = 4$ elements, and $n \in \{2, \ldots, 6, 9\}$ strategies. We use our own C++ implementation of more than 10,000 lines of code for the generation of random instances, the binary search scheme, and most of the experimentation framework; post-processing of the gained data is done using the R System [74]. Solvers for general NLPs, namely Ipopt [86, 89] and Lancelot [31], are used to treat the involved NLPs. We use a simple test to determine whether the solution returned by the NLP solver is in fact a Nash equilibrium, which may not be the case,

e.g., if only a local optimum is found. However, for the majority of cases, at least one Nash equilibrium was found, and we could in some cases compute a price of anarchy up to the best known theoretical lower bound (which is a factor of γ away from the upper bound). This points to an ability of our algorithm to find worst-case equilibria. We call the ratio between the social cost of a Nash equilibrium with maximum social cost and the optimum, both emanating from our computations, the *observed price of anarchy*. If the computation in fact yields a worst-case Nash equilibrium and an optimum, the observed price of anarchy equals the price of anarchy. The observed price of anarchy is a lower bound on the price of anarchy in general.

It is notable that our study of more than 10 million random instances yielded several ones with an observed price of anarchy equal to the theoretical lower bound – but never above it (up to inaccuracies). This justifies to raise the conjecture that the theoretical lower bound is in fact an upper bound.

We also use the experiments to compare our bounds – the proven and the conjectured one – to another type of bound, introduced by Perakis [72], which is based on the operator norm of a matrix related to the Jacobian of the strategy latency function (as a function $\mathbb{R}^n_{\geq 0} \longrightarrow \mathbb{R}^n_{\geq 0}$, where n is the number of strategies). It is only applicable if the Jacobian is positive (semi-)definite, which may or may not be the case for NCRCGs. For those instances where Perakis' bound is applicable, we compare it to our bounds. Both approaches show their own strengths and weaknesses: our bounds are computationally much simpler to obtain than Perakis'; and indeed, at least our implementation for computing Perakis' bound yielded some extraordinarily high values, of which it is not known whether they should be attributed to numerical instabilities or represent the true values. On the other hand, for most cases, Perakis' bound is smaller than ours, and hence better. However, the margin is considerably smaller for our conjectured bound than for our proven bound. Our findings accentuate the importance of developing multiple structurally different approaches for computing bounds on the price of anarchy.

Part II. The main achievements in the second part are lower and upper bounds on the price of anarchy in distributed network formation when the network is subject to modification by an adversary. We recall that in our framework of distributed network formation each vertex represents a player and each player can build links to other players. Links cost an amount of α each, but can also have a positive effect by improving connectivity.

In our model, an adversary is allowed to delete one link after the network is built. The adversary is modeled by a random experiment, more precisely: by a probability distribution on the links of the network that was built. This probability distribution may even depend on that network, e.g., the adversary might favor links that separate a maximum number of vertex pairs. The individual cost function of a vertex v incorporates the expected number of vertices to which v will become disconnected when the adversary strikes.

We study the unilateral case using Nash equilibrium, and the bilateral case using pairwise Nash equilibrium and pairwise stability. Two types of adversaries are analyzed: a simple-minded one, adhering to the uniform probability distribution, and a smart one, which chooses the link to destroy with the objective of separating a maximum number of vertex pairs. For the unilateral case and both adversaries, we show a price of stability of $1 + o(1)$ and – as one of the main results – an $O(1)$ bound on the price of anarchy. Bounding building cost in an equilibrium for both adversaries works by recognizing that equilibrium graphs are chord-free. Bounding indirect cost, (i.e., the cost expressing connectivity properties of the built network) works differently for both adversaries. For the simple-minded adversary, we use a diameter argument, roughly similar to those found for other models in existing literature, e.g., for the sum-distance model [41]. For the smart adversary, a new approach has to be taken: we look at the macroscopic structure of the network, represented by a tree, and distinguish several ways in which vertices are distributed across subtrees.

For the bilateral case, the adversaries show different characteristics: for the simple-minded one we show a bound of $O(1 + \sqrt{n/\alpha})$ on the price of anarchy; we do not know whether it is tight. For the smart adversary, we show a lower bound of $\Omega(n)$ for $\alpha > 2$ considered constant,

which is the worst that can happen for *any* adversary in this model if $\alpha = \Omega(1)$. This provides an example that unilateral and bilateral network formation can behave substantially different.

We also study convexity[4] of individual cost functions. This is important for the relation between pairwise Nash equilibrium and pairwise stability: both equilibrium concepts are equivalent under convexity of individual cost functions. We show that for the simple-minded adversary, individual cost functions are convex. For the smart adversary, we provide an example that they are not convex – however, we only succeed in showing this outside the set of pairwise stable networks. We do not yet know whether pairwise Nash equilibrium and pairwise stability are equivalent for the smart adversary. The example still provides indication that network formation with a smart adversary is in some sense substantially different from the known sum-distance model [41] or our model with a simple-minded adversary, since these have individual cost functions that are everywhere convex.

[4]This is a different convexity notion than the one we encounter in the first part.

Future Work

Our results raise several questions that we expect to be a driving force for future research. Can we prove the conjecture for the bound on the price of anarchy in NCRCGs? Can better algorithms be devised for computation of extreme equilibria? Our experimental findings can serve as a benchmark for future approaches: will a new algorithm deliver substantially higher observed prices of anarchy (for the same random model)? More directions for future work on NCRCGs are listed on p. 90.

Regarding network formation, we leave open as one of the most intriguing questions how the situation changes when an adversary is allowed to delete more than one link. Our proofs rely heavily on the fact that only one link is deleted, so we expect this to be a new challenge. Moreover, we pointed out substantial differences between the unilateral and bilateral case. This is especially interesting since comparison of unilateral and bilateral network formation under the aspect of the price of anarchy has received much attention recently, e.g., [32, 38], and models that combine aspects of both principles are beginning to be discussed [14]. It would be interesting to study our adversary model in such a new setting. More directions for future work on network formation are listed on p. 165.

Part I

Distributed Non-Atomic Routing in Networks

Chapter 1

Non-Atomic Congestion Games

We briefly review previous work in this chapter. Everything is in a non-atomic model, even when the term "non-atomic" is omitted sometimes. That is, each player has only a negligible influence on the system; it will be elaborated in detail what this means exactly. Our starting point is selfish unicast routing. A large number of players wishes to send a small amount of data each from one source to one destination through a communication network. Each player decides on her own which route to take, aiming to optimize the individual transmission time. How is traffic expected to be distributed across available routes? How efficient is the outcome?

We take an incremental approach for the presentation. In Sec. 1.1, we consider unicast routing, mostly as a motivation. We then turn to non-atomic congestion games (NCG), introduced in Sec. 1.2 and Sec. 1.3. NCGs are one of the most general previously studied models. They encompass unicast routing, but they do *not* fully encompass multicast routing, that is, when traffic flows from each source to multiple terminals at the same time. We will introduce a new model for different kinds of multicast in Ch. 2. In Sec. 1.4 we elaborate on the non-atomic equilibrium concept from a more general perspective; this section is particularly important for the rest of Part I. In Sec. 1.5 to Sec. 1.7 we present known results on NCGs, with a special focus on the price of anarchy, among others the results and proofs due to Roughgarden and Tardos as well as Correa, Schulz, and Stier Moses. The essence is that we can tightly bound the price of anarchy with a single parameter.

Particularly important for the rest of Part I is Sec. 1.7.2, which introduces the β parameter. This parameter will be helpful for the analysis of non-atomic consumption-relevance congestion games (and selfish multicast routing) in Ch. 2.

In Sec. 1.8 we take a step back and look at the model under the aspect of what reasons are responsible for inefficiency of equilibria. Both principles – "absence of central authority" and "lack of cooperation" – will be recognized. We give a short chronological overview over existing literature in Sec. 1.9.

1.1 Selfish Unicast Routing

We start with a simple example, originally due to Pigou [73]. Consider two cities s and t, connected by two highways H_1 and H_2. Many commuters have to travel from s to t at the same time, and each of them can choose between H_1 and H_2. Highway H_1 is rather long, but broad enough so that no congestion effects occur: no matter how many commuters choose highway H_1, their travel time – the latency – will always be one hour. Highway H_2 is much shorter, but also narrower. Commute time on H_2 is influenced by congestion effects: the time grows proportional to the number of drivers choosing H_2, and it equals one hour if all drivers choose H_2 (and none chooses H_1). Let us normalize the total amount of drivers to 1, and say that the commute time experienced by drivers on H_2 takes x times one hour if x drivers take that highway, $x \in [0,1]$.

If $x < 1$, then the drivers on H_2 experience less commute time than those on H_1, and so we expect that as long as $x < 1$, the $1 - x$ drivers on H_1 see an incentive in switching to H_2. It follows that we have an equilibrium if and only if $x = 1$. (We will give a formal definition of equilibrium below.) As social cost we use the total travel time, which is $1 \cdot (1 - x) + x \cdot x = 1 - x + x^2$. It is minimum, namely $\mathsf{OPT} = \frac{3}{4}$, when $x = \frac{1}{2}$; it is 1 when $x = 1$, i.e., in equilibrium. The price of anarchy hence is $\frac{4}{3}$. If a central authority for minimizing the total travel time was in charge, that authority would enforce $x = \frac{1}{2}$, i.e., one half of the drivers would be allowed to take highway H_2, and the other half would be *forced* to take highway H_1 and accept that those on highway H_2 are better off.

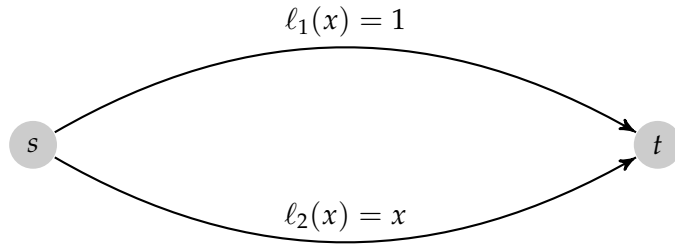


Figure 1.1. Pigou's example.

More generally, let $G = (V, E)$ be a directed multigraph. Let there be N source-terminal pairs $s_i, t_i \in V$, $i \in [N]$, and for each a demand $d_i > 0$ that is to be routed from s_i to t_i. Each edge $e \in E$ has a real-valued function $\ell_e : \mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}_{\geq 0}$, called *edge latency function*. We usually assume each edge latency function to be continuous and non-decreasing. When an amount of x is routed through edge e, it takes $\ell_e(x)$ units of time per unit, hence $\ell_e(x)x$ in total. Let $\mathcal{P}_i$ be the set of all s_i-t_i paths that shall be eligible for routing d_i, and assume for simplicity that $\mathcal{P}_i \cap \mathcal{P}_j = \emptyset$ for all $i, j \in [N]$, $i \neq j$. Set $\mathcal{P} := \dot{\bigcup}_{i \in [N]} \mathcal{P}_i$ and denote for each $e \in E$ by $\mathcal{P}(e)$ those paths in $\mathcal{P}$ that traverse e. A *flow* f is a vector $(f_P)_{P \in \mathcal{P}}$ of non-negative real numbers such that $\sum_{P \in \mathcal{P}_i} f_P = d_i$ for all $i \in [N]$. Let f be a flow and fix an edge $e \in E$. The total flow through e is called its *congestion*, denoted $g_e(f)$, i.e., $g_e(f) := \sum_{P \in \mathcal{P}(e)} f_P$. Routing one unit of flow along a path $P \in \mathcal{P}$ takes time $L_P(f) := \sum_{e \in E(P)} \ell_e(g_e(f))$; we call this the *path latency*. The total time for routing all the demands is $\mathsf{SC}(f) := \sum_{P \in \mathcal{P}} L_P(f) f_P$; we call this the *social cost*. Fig. 1.1 shows how Pigou's example looks using this formalism; we have $N = 1$ and $d_1 = 1$.

Now we introduce the distributed aspect. Fix $i \in [N]$. Routing of d_i from s_i to t_i shall not be subject to a central control, but it rather shall be controlled by a (large) number of individuals, which we will call *players*. We assume a large number of players, of which each only controls a negligible fraction of the demand. An illustrative example for this is road traffic. The decision of a single driver which route to take will have *virtually no effect* on congestions and latencies of the road network. Another example is a communication network with a large

number of users where each user has only got a small amount of data to send or receive.

So, a flow f gives a distribution of players across available paths. All players that have chosen a path $P \in \mathcal{P}$ experience latency $L_P(f)$; this is their individual cost. If there are $P, Q \in \mathcal{P}_i$ such that $L_P(f) < L_Q(f)$, then the fraction f_Q of the players that is on path Q has an incentive of switching from path Q to path P. We choose as equilibrium concept the *Nash equilibrium*, that is, in an equilibrium no player shall have an incentive of switching to another path, given the decisions of the other players. This is formalized as follows. We say that flow f is a Nash equilibrium if

$$\forall i \in [N] \quad \forall P, Q \in \mathcal{P}_i : \quad \big(f_P > 0 \implies L_P(f) \leq L_Q(f) \big) \ .$$

In other words, we have an equilibrium if and only if all flow travels along minimum-latency paths. We will elaborate more on this equilibrium concept and on existence of Nash equilibria in Sec. 1.4.

Routing in a distributed way is also known as *selfish routing*. We speak of *selfish unicast routing* when emphasizing that routing goes from each source to exactly one terminal each (but there can be multiple source-terminal pairs). A comprehensive treatment of selfish unicast routing can be found in Roughgarden's dissertation [77]. Recall that we restrict to non-atomic models, while selfish routing is also of concern in other settings; see, e.g., the survey by Czumaj [35]. We will state the most important known results on (non-atomic) selfish routing in the context of the more general non-atomic congestion games (NCG) in Sec. 1.5 to Sec. 1.7.

Concluding this introduction, we study a generalization of Pigou's example with a parameter $p \in \mathbb{N}_{\geq 1}$. The latency function $\ell_2(x) = x$ is exchanged for $\ell_2(x) = x^p$. As before, we denote the upper path by H_1 and the lower path by H_2, both consisting of only one edge each. The edges are denoted 1 and 2. We compute the path latencies and the social cost for some arbitrary flow $f = (f_{H_1}, f_{H_2})$. We have $L_{H_1}(f) = \ell_1(g_1(f)) = 1$ and $L_{H_2}(f) = \ell_2(g_2(f)) = \ell_2(f_{H_2}) = f_{H_2}^p$ and

$$\mathsf{SC}(f) = L_{H_1}(f) f_{H_1} + L_{H_2}(f) f_{H_2} = f_{H_1} + f_{H_2}^{p+1} \overset{f_{H_1}+f_{H_2}=1}{=} 1 - f_{H_2} + f_{H_2}^{p+1} \ .$$

We see immediately that the only Nash equilibrium is $f \coloneqq (0,1)$, for if any fraction of the flow was to take the upper path, say, $f' \coloneqq (\varepsilon, 1-\varepsilon)$, we would have $L_{H_2}(f') = (1-\varepsilon)^p < 1 = L_{H_1}(f')$, meaning that not all flow would travel along minimum-latency paths. On the other hand, simple real calculus gives that $f^* \coloneqq (1-(p+1)^{-1/p}, (p+1)^{-1/p})$ is the only optimal flow. Its social cost is

$$\begin{aligned}
\mathsf{SC}(f^*) &= 1 - (p+1)^{-1/p} + (p+1)^{-(p+1)/p} \\
&= 1 - \left((p+1)\,(p+1)^{-1-\frac{1}{p}} - (p+1)^{-(p+1)/p}\right) \\
&= 1 - \Big(p\,(p+1)^{-1-\frac{1}{p}} + \underbrace{(p+1)^{-1-\frac{1}{p}} - (p+1)^{-(p+1)/p}}_{=0}\Big) \qquad (1.1) \\
&= 1 - p\,(p+1)^{-1-\frac{1}{p}} \ .
\end{aligned}$$

Hence, the price of anarchy is $\frac{\mathsf{SC}(f)}{\mathsf{SC}(f^*)} = (1 - p\,(p+1)^{-1-\frac{1}{p}})^{-1} = \Theta(\frac{p}{\ln p})$, which (slowly) tends to ∞ as p tends to ∞. Pigou's example allows two observations: first, we see that arbitrary high prices of anarchy can be achieved with very simple networks – just one source, one terminal, and parallel edges in between. Second, we observe a correlation between the price of anarchy and the kind of latency functions used – the higher the degree of the polynomial ℓ_2, the larger the price of anarchy becomes. In fact, Roughgarden [77] shows that the price of anarchy in unicast routing can be upper-bounded by a value only dependent on the class $\mathcal{L}$ of eligible edge latency functions, the so-called *anarchy value* $\alpha(\mathcal{L})$, provided that $\mathcal{L}$ fulfills some moderate conditions. For affine element latency functions, as in Pigou's example with $p = 1$, the anarchy value is $\frac{4}{3}$, and hence Pigou's example marks the worst case. We will elaborate on this in Sec. 1.7.

1.2 Non-Atomic Congestion Games

Many interesting results on the price of anarchy in selfish unicast routing actually do not rely on a graph structure, but can be stated in a more general context, described in the following. We have a set of elementary resources, called *elements*, which can only be utilized in

certain combinations, called *strategies*. Players have to choose one or more strategies to accomplish certain tasks; in a non-atomic model, a player can only choose one strategy. Each element performs its part of the task with a latency. This latency is experienced by those players who utilize the respective element via some strategy. Players try to avoid long latencies – they want their tasks completed as soon as possible. In unicast routing, the elementary resources are edges in the network and strategies are paths. However, such a network-oriented structure is not necessary for many results to hold.

1.1 Definition. An instance in the *non-atomic congestion game* model, shortly referred to as "an NCG", is defined by the following five items:

(i) A set E of *elementary resources* or *elements*, say $E = [m]$ for an $m \in \mathbb{N}$.

(ii) For each $e \in E$ a real, continuous, non-decreasing function $\ell_e : \mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}_{\geq 0}$, called the *element latency function* of e.

(iii) Non-empty subsets $\mathfrak{S} \subseteq 2^E$ of E, called *strategies*. Denote $n := |\mathfrak{S}|$ and for each $e \in E$ denote the set of strategies which contain e by $\mathfrak{S}(e) := \{S \in \mathfrak{S};\ e \in S\}$.

(iv) For each $e \in E$ and $S \in \mathfrak{S}$ a positive real number r_{eS}, called *rate of consumption* (of e under S).

(v) Numbers $d_1, \ldots, d_N \in \mathbb{R}_{>0}$, called *demands*, and a partition $\mathfrak{S} = \dot{\bigcup}_{i \in [N]} \mathfrak{S}_i$. We call each $i \in [N]$ a *player class*. So each player class i has a demand d_i and some strategies $\mathfrak{S}_i$; denote $n_i := |\mathfrak{S}_i|$.

We also use the term *element latency function* to refer to any continuous, non-decreasing function $\mathbb{R} \longrightarrow \mathbb{R}$.

All the following notions are relative to a fixed NCG. An *action distribution* is a vector $a \in \mathbb{R}^n_{\geq 0}$ such that $\sum_{S \in \mathfrak{S}_i} a_S = d_i$ for all $i \in [N]$. So, an action distribution specifies a way of distributing the demand of each player class across available strategies. Let $\mathcal{A}$ be the set of all action distributions. We define *congestion*, *strategy latency*, and *social cost*, respectively:

$$g_e(a) := \sum_{S \in \mathfrak{S}(e)} r_{eS} a_S \qquad \text{for } a \in \mathcal{A} \text{ and } e \in E$$

$$L_S(a) \coloneqq \sum_{e \in S} r_{eS} \ell_e(g_e(a)) \qquad \text{for } a \in \mathcal{A} \text{ and } S \in \mathfrak{S}$$
$$\mathsf{SC}(a) \coloneqq \sum_{S \in \mathfrak{S}} L_S(a) a_S \qquad \text{for } a \in \mathcal{A}$$

For an action distribution $a \in \mathcal{A}$ and element $e \in E$ we call $\ell_e(g_e(a))$ the *element latency* of e under a. Denote also for an action distribution $a \in \mathcal{A}$ the following vectors:

$$\vec{g}(a) \coloneqq (g_e(a))_{e \in E} \in \mathbb{R}^m_{\geq 0}$$
$$\vec{\ell}(a) \coloneqq (\ell_e(g_e(a))_{e \in E} \in \mathbb{R}^m_{\geq 0}$$
$$\vec{L}(a) \coloneqq (L_S(a))_{S \in \mathfrak{S}} \in \mathbb{R}^n_{\geq 0}$$

Then the social cost can be written as a scalar product $\mathsf{SC}(a) = \vec{L}(a)^\top a$.

Having the rates of consumption in congestions as well as in strategy latencies may appear an arbitrary decision at first. In Sec. 1.6 and Sec. 1.7, however, we will see that it is essential. It allows us to express the social cost on strategy level as well as on element level, namely we have

$$\mathsf{SC}(a) = \sum_{S \in \mathfrak{S}} L_S(a) a_S = \sum_{e \in E} \ell_e(g_e(a)) g_e(a) \ . \tag{1.2}$$

1.2 Definition.

(i) An action distribution $a \in \mathcal{A}$ is called a *Nash equilibrium*, abbreviated "N.E.", if

$$\forall i \in [N] \quad \forall S, T \in \mathfrak{S}_i : \quad \big(a_S > 0 \implies L_S(a) \leq L_T(a) \big) \ .$$

(ii) An action distribution $a^* \in \mathcal{A}$ is called *optimal*, or an *optimum*, if

$$\mathsf{SC}(a^*) = \min_{a \in \mathcal{A}} \mathsf{SC}(a) \ .$$

Denote $\mathsf{OPT} \coloneqq \mathsf{SC}(a^*)$ for an optimum a^*.

Nash equilibria always exist in this model, as we will see in Sec. 1.4. By continuity of $\vec{L}$ and compactness of $\mathcal{A}$, optima always exist as well. We only consider the case $\mathsf{OPT} > 0$. This is not a limiting assumption. For, if there was an action distribution a with $\mathsf{SC}(a) = 0$, then $L_S(a) = 0$

for all $S \in \mathfrak{S}$ such that $a_S > 0$. Hence a would be a Nash equilibrium. We will see later that all Nash equilibria have the same social cost. So all Nash equilibria would be optimal in this special case, and hence distributed operation would have no negative effect on the social cost. It also becomes clear now why we do not allow empty sets as strategies: an empty strategy would provide a 'free ride' for everyone in the respective player class. That would make the player class irrelevant, and even lead to $\mathsf{OPT} = 0$ if there are no other player classes.

1.3 Definition. We define the *price of stability* and the *price of anarchy* of an NCG, respectively, by

$$\inf_{\substack{a \in \mathcal{A} \\ a \text{ is N.E.}}} \frac{\mathsf{SC}(a)}{\mathsf{OPT}} \qquad \text{and} \qquad \sup_{\substack{a \in \mathcal{A} \\ a \text{ is N.E.}}} \frac{\mathsf{SC}(a)}{\mathsf{OPT}} \ .$$

By continuity, we can replace the infimum and supremum by minimum and maximum, respectively. We will see in Sec. 1.5 that price of stability and anarchy coincide in NCGs. They may, however, diverge in the NCRCG model studied in Ch. 2 and Ch. 3.

To describe a unicast routing game as an NCG, we simply let network links correspond to elements, paths to strategies (a path is fully specified by the set of its links), and source-terminal pairs to player classes. The rates of consumption are chosen $r_{eS} = 1$ for all $S \in \mathfrak{S}$ and $e \in S$. So, the NCG model encompasses the unicast routing model.

NCGs are useful since they are more general. By an appropriate setting of the rates of consumption, we can model two forms of multicast routing with NCGs, as shown in Ch. 2. NCGs are, however, not general enough to encompass certain other interesting forms of multicast routing. This is where the NCRCG model comes in, also presented in Ch. 2.

1.3 Formal Subtleties – Matrix Notation

We have defined the set of strategies as a *subset* of 2^E. This gives a good intuition, but it is also an unnecessary restriction. For, it does not allow the same strategy being shared by two player classes (unless we renounce that $\mathfrak{S}_1, \ldots, \mathfrak{S}_N$ form a partition, which would cause notational inconvenience in other places). The restriction also prevents

us from having two 'strategies' on the same elements but with different rates of consumption.

The common approach in the literature is to let $\mathfrak{S}$ be a "multiset". This does not address the matter, however, since a multiset is just a set where elements have multiplicities; it does not provide means to refer to a particular 'copy' of the same element. A solution is to have a mapping $\iota : [n] \longrightarrow 2^E$ and to always refer to strategies by their pre-image under ι, conveniently also referred to as *index*. When using a notation such as $S_j \coloneqq \iota(j)$, $j \in [n]$, we also speak of an *indexed family*.

Another solution uses a matrix; this is the one we prefer. It allows us to specify an NCG in the following way. Let

$$(r_{eS})_{\substack{e=1,\dots,m \\ S=1,\dots,n}} \in \mathbb{R}_{\geq 0}^{m \times n}$$

be a matrix. We call its entries *rates of consumption*. We use letters "e" and "S" for row and column indices, respectively, in order to remind us that these indices point to something that conceptually is an element or a strategy for us. Elements correspond to rows and strategies correspond to columns; actually we continue speaking of *elements* and *strategies*. Positive entries in column S mark elements that are included in strategy S in the sense of the original notation. In particular there is a positive entry in each column by our requirement that strategies are non-empty. We have *element latency functions* as usual, one for each element (or, in other words, for each row of the matrix). We denote $\mathfrak{S} \coloneqq [n]$ and have a partition $\mathfrak{S} = \biguplus_{i \in [N]} \mathfrak{S}_i$ to express different *player classes*. This means that $\mathfrak{S}_i$ contains the indices of columns in the matrix that describe strategies available to player class i. There are positive *demands* $d_1, \dots, d_N$ as usual, one for each player class.

The matrix notation makes congestions and strategy latencies notationally simpler: we can always sum over *all* strategies or elements, respectively:

$$g_e(a) \coloneqq \sum_{S \in \mathfrak{S}} r_{eS} a_S \qquad \text{for } a \in \mathcal{A} \text{ and } e \in E$$

$$L_S(a) \coloneqq \sum_{e \in E} r_{eS} \ell_e(g_e(a)) \qquad \text{for } a \in \mathcal{A} \text{ and } S \in \mathfrak{S}$$

We will assume the matrix notation from now on.

1.4 Non-Atomic Equilibrium Concept

1.4.1 Basic Concept

Recall the equilibrium definition from Def. 1.2(i). It expresses that all the demand is distributed across strategies with minimum latency. We explain in more detail why this describes an equilibrium. More precisely, we explain why this fits the notion of Nash equilibrium, which is a fundamental equilibrium concept and reads: *no player has an incentive to switch strategies, given that all other players stick to their current strategies.* An incentive to switch strategies would be that the new strategy offers strictly lesser individual cost. In a non-atomic model, each player can only choose one strategy. In the NCG model, the individual cost is the strategy latency. An incentive to switch from strategy S to T would be that T offers strictly lesser latency than S. Since a single player is assumed to have only a negligible impact on the system (i.e., on congestions and latencies), the determination whether strategy T offers a strictly lesser latency than S is easily done: simply check whether $L_S(a) < L_T(a)$ for the current action distribution a. We do not have to take into account whether the change from S to T of that single player in question might influence the strategy latencies – we know, by assumption, that a single player does not change the system. Actually, we never defined what a 'single player' is. Instead, we subsume all players in class i to the real interval $[0, d_i]$. An action distribution, from this perspective, defines a partition for each of those intervals. Nash equilibria in such a model are also known as *Wardrop equilibria*, since the specific concept expressed in Def. 1.2(i) is due to Wardrop [87]. We quote [87, p. 344–345], emphasis added:

> The problem is to discover how traffic may be expected to distribute itself over alternative routes, and whether the distribution adopted is the most efficient one. [...]
>
> Consider two alternative criteria [...] which can be used to determine the distribution on the routes, as follows:
>
> (1) **The journey times on all the routes actually used are equal, and less than those that would be experienced by a single vehicle on any unused route.**

(2) The average journey time is a minimum.

The first criterion is quite a likely one in practice, **since it might be assumed that traffic will tend to settle down into an equilibrium situation in which no driver can reduce his journey time by choosing a new route.** On the other hand, the second criterion is the most efficient in the sense that it minimizes the vehicle-hours spent on the journey.

Our definition of Nash equilibrium matches that first criterion.

1.4.2 Non-Atomic Games and Variational Inequality Formulation

We show how Nash equilibria in an NCG can be characterized by a *variational inequality*. This result in fact goes far beyond NCGs: it only uses the notion of strategy latency and does not require that the relations between strategies are described via elementary resources or rates of consumption. We will use this result throughout the first part of this thesis, also for more general games than NCGs. Therefore, we make the definition of a *non-atomic game*. It is a simplification of the concept known under the same name in the literature, but it suffices for our purposes.

1.4 Definition. An instance of a *non-atomic game* is defined by the following items:

(i) A set $\mathfrak{S}$, say $\mathfrak{S} = [n]$ for a number $n \in \mathbb{N}$.

(ii) Numbers $d_1, \ldots, d_N \in \mathbb{R}_{>0}$ and a partition $\mathfrak{S} = \biguplus_{i \in [N]} \mathfrak{S}_i$. This defines the set of action distributions $\mathcal{A}$ as for NCGs.

(iii) For each $S \in \mathfrak{S}$ a function $L_S : U \longrightarrow \mathbb{R}_{\geq 0}$ such that $\mathcal{A} \subseteq U \subseteq \mathbb{R}^n$.

Social cost, optima, and Nash equilibria are defined as for NCGs. Often, $\vec{L}$ is given as a function $\mathbb{R}^n \longrightarrow \mathbb{R}^n$. Let

$$\Lambda_i(a) \coloneqq \min_{S \in \mathfrak{S}_i} L_S(a)$$

for $a \in \mathcal{A}$ and $i \in [N]$. It follows directly from the definition:

1.5 Proposition. *For a non-atomic game, an action distribution $a \in \mathcal{A}$ is a Nash equilibrium if and only if*

$$\forall i \in [N] \quad \forall S \in \mathfrak{S}_i : \quad \big(a_S > 0 \implies L_S(a) = \Lambda_i(a) \big) \ .$$

Moreover, the social cost of a Nash equilibrium a is

$$\mathsf{SC}(a) = \sum_{i \in [N]} \Lambda_i(a) d_i \ . \tag{1.3}$$

For the following characterization of Nash equilibria, we need the *mixed social cost*, that is similar to the social cost, but dependent on *two* action distributions:

$$\mathsf{SC}^a(b) := \sum_{S \in \mathfrak{S}} L_S(a) b_S \qquad \text{for } a, b \in \mathcal{A}.$$

The following theorem is due to Smith [84]. It will be crucial in the following.

1.6 Theorem. *For a non-atomic game, an action distribution $a \in \mathcal{A}$ is a Nash equilibrium if and only if*

$$\mathsf{SC}(a) \le \mathsf{SC}^a(b) \quad \textit{for all } b \in \mathcal{A}. \tag{VAR}$$

Proof. Let first a be a Nash equilibrium. Fix any $b \in \mathcal{A}$. By Prop. 1.5 we have

$$\begin{aligned} \mathsf{SC}^a(b) - \mathsf{SC}(a) &= \sum_{i \in [N]} \sum_{S \in \mathfrak{S}_i} \underbrace{L_S(a)}_{\ge \Lambda_i(a)} b_S - \sum_{i \in [N]} \Lambda_i(a) d_i \\ &\ge \sum_{i \in [N]} (\Lambda_i(a) \underbrace{\sum_{S \in \mathfrak{S}_i} b_S}_{= d_i} - \Lambda_i(a) d_i) = 0 \ . \end{aligned}$$

Now, let $a \in \mathcal{A}$ such that (VAR) holds. Fix $i_0 \in [N]$ and pick $S_0 \in \mathfrak{S}_{i_0}$ such that $L_{S_0}(a) = \Lambda_{i_0}(a) =: \Lambda$. Define $b \in \mathbb{R}^n_{\ge 0}$ by

$$b_S := \begin{cases} d_{i_0} & \text{if } S = S_0 \\ 0 & \text{if } S \in \mathfrak{S}_{i_0} \setminus \{S_0\} \\ a_S & \text{otherwise} \end{cases} .$$

That is, we shift all the demand for i_0 to strategy S_0. Then b is an action distribution, and by (VAR) we have

$$\begin{aligned} 0 &\leq \sum_{i\in[N]} \sum_{s\in\mathfrak{S}_i} L_S(a) \underbrace{(b_S - a_S)}_{=0 \text{ if } i\neq i_0} = \sum_{S\in\mathfrak{S}_{i_0}} L_S(a)\,(b_S - a_S) \\ &= \underbrace{L_{S_0}(a)}_{=\Lambda} d_{i_0} - \sum_{S\in\mathfrak{S}_{i_0}} \underbrace{L_S(a)}_{\geq\Lambda} a_S \ . \end{aligned}$$

Since $\sum_{S\in\mathfrak{S}_{i_0}} a_S = d_{i_0}$, it follows that $L_S(a) = \Lambda$ whenever $a_S > 0$, $S \in \mathfrak{S}_{i_0}$. Because i_0 was taken arbitrarily, a is a Nash equilibrium. □

1.7 Theorem. *Every non-atomic game for which L_S is continuous for each $S \in \mathfrak{S}$ admits a Nash equilibrium.*

Proof. Equation (VAR) is a well-studied variational inequality. We know that solutions exist under the continuity assumption on the strategy latency functions since $\mathcal{A}$ is compact and convex, see, e.g., [68, Thm. 1.4] or [52, Thm. 3.1]. □

Since we require element latency functions to be continuous in NCGs, we receive immediately:

1.8 Corollary. *Every NCG admits a Nash equilibrium.*

Condition (VAR) has an interesting interpretation for the non-atomic equilibrium concept. The mixed social cost $\mathsf{SC}^a(b)$ gives us a kind of 'social cost' dependent on two parameters: the first one, namely a, defines the *system* (i.e., congestions and corresponding element and strategy latencies), and the second one, namely b, indicates how that system is used. Condition (VAR) says that Nash equilibria are exactly those action distributions a that give the optimal way to use the system that is already fixed by a.

1.5 Uniqueness of Nash Equilibrium

We consider NCGs again. Nash equilibria are not unique in general – just consider constant latency functions. However, all Nash equilibria have the same element latencies and social cost, and if element latency functions are strictly increasing, we have unique congestions.

1.9 Theorem. *Let $a, a' \in \mathcal{A}$ be Nash equilibria. Then*

(i) $\ell_e(g_e(a)) = \ell_e(g_e(a'))$ *for all* $e \in E$, *and hence* $g_e(a) = g_e(a')$ *if* ℓ_e *is strictly increasing;*

(ii) $\mathsf{SC}(a) = \mathsf{SC}(a')$.

Proof. By Thm. 1.6, $\mathsf{SC}(a) - \mathsf{SC}^a(a') \leq 0$ and $\mathsf{SC}(a') - \mathsf{SC}^{a'}(a) \leq 0$, so

$$\begin{aligned}
0 &\geq \mathsf{SC}(a) - \mathsf{SC}^a(a') + \mathsf{SC}(a') - \mathsf{SC}^{a'}(a) \\
&= \sum_{S \in \mathfrak{S}} (L_S(a) - L_S(a'))\,(a_S - a'_S) \\
&= \sum_{S \in \mathfrak{S}} \sum_{e \in E} r_{eS}\,(\ell_e(g_e(a)) - \ell_e(g_e(a')))\,(a_S - a'_S) \\
&= \sum_{e \in E} (\ell_e(g_e(a)) - \ell_e(g_e(a'))) \sum_{S \in \mathfrak{S}} r_{eS}(a_S - a'_S) \\
&= \sum_{e \in E} (\ell_e(g_e(a)) - \ell_e(g_e(a')))\,(g_e(a) - g_e(a'))\ .
\end{aligned}$$

The last equation was positive if we had $\ell_e(g_e(a)) \neq \ell_e(g_e(a'))$ for some e; recall that element latency functions are non-decreasing. This shows (i). It follows furthermore that $L_S(a) = L_S(a')$ for all $S \in \mathfrak{S}$ and also that $\Lambda_i(a) = \Lambda_i(a')$ for all $i \in [N]$. Assertion (ii) now follows with Prop. 1.5. □

Note how important it is for the proof that we have rates of consumption in strategy latencies as well as in congestions. This allows us to draw a bow from strategy latencies, about which we have some knowledge by the variational inequality, to congestions and element latencies, for which we want to prove something. We conclude with a simple observation following from the previous theorem.

1.10 Corollary. *The price of stability and price of anarchy coincide in NCGs.*

1.6 Computation and Potential Function

1.11 Definition.

(i) An element latency function ℓ is called *standard* if it is continuously differentiable[1] and $x \mapsto \ell(x)x$ is convex.

[1] The original definition by Roughgarden [77, Def. 2.3.5] only demands differen-

(ii) A class of element latency functions $\mathcal{L}$ is called *standard* if each $\ell \in \mathcal{L}$ is standard and $\mathcal{L}$ contains at least one non-zero function.

(iii) For each $p \in \mathbb{N}_{\geq 1}$ let $\mathrm{Poly}^+(p)$ be the class of polynomials of degree at most p and non-negative coefficients. These classes are clearly standard.

We start with the computation of optima. Optima are characterized by the following non-linear program.

$$\begin{aligned} \text{minimize} \quad & \sum_{e \in E} \ell_e(x_e) x_e \\ \text{subject to} \quad & x_e - \sum_{S \in \mathfrak{S}} r_{eS} a_S = 0 \quad \forall\, e \in E \\ & d_i - \sum_{S \in \mathfrak{S}_i} a_S = 0 \quad \forall\, i \in [N] \\ & -a_S \leq 0 \quad \forall\, S \in \mathfrak{S} \end{aligned} \tag{OPT NLP}$$

The constraints ensure that for any feasible $(x, a) = ((x_e)_{e \in E}, (a_S)_{S \in \mathfrak{S}})$ we have that $a \in \mathcal{A}$ and $x = \vec{g}(a)$.

The set of action distributions $\mathcal{A}$ is convex, and so is the function SC, if, e.g., each $x \mapsto \ell_e(x)x$ is convex, as it is the case for standard element latency functions. Hence, for standard element latency functions, optima are characterized by a convex program. Convex programs often are computationally convenient; we will make some more remarks on this in Sec. 2.11.

To compute Nash equilibria, we define the potential function. Let

$$\hat{\ell}_e : \mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}_{\geq 0}, x \mapsto \int_0^x \ell_e(t)dt \quad \text{for each } e \in E.$$

Then each $\hat{\ell}_e$ is differentiable on $\mathbb{R}_{\geq 0}$ and can be extended straightforwardly to a differentiable function on $\mathbb{R}$, simply set

$$\hat{\ell}_e(x) \coloneqq \left(\frac{\mathrm{d}}{\mathrm{d}y}\hat{\ell}_e(y)\right)(0)\, x = \ell_e(0)\, x$$

tiability. However, we will need a continuous derivative later when we consider marginal cost functions.

for $x \in \mathbb{R}_{<0}$. We assume this extension in the following. Define

$$\Phi : \mathbb{R}^m \longrightarrow \mathbb{R}, (x_e)_{e\in E} \mapsto \sum_{e\in E} \hat{\ell}_e(x_e) \ .$$

We call Φ the *potential*; note that $\nabla\Phi = \vec{\ell}$ on $\mathbb{R}^m_{\geq 0}$. Since each ℓ_e is non-decreasing, each $\hat{\ell}_e$ and the potential Φ are all convex, and hence the following is a convex program (not only for standard element latency functions).

$$\begin{aligned} \text{minimize} \quad & \Phi((x_e)_{e\in E}) \quad (= \sum_{e\in E} \hat{\ell}_e(x_e)) \\ \text{subject to} \quad & x_e - \sum_{S\in\mathfrak{S}} r_{eS} a_S = 0 \quad \forall\, e \in E \\ & d_i - \sum_{S\in\mathfrak{S}_i} a_S = 0 \quad \forall\, i \in [N] \\ & -a_S \leq 0 \quad \forall\, S \in \mathfrak{S} \end{aligned} \tag{Nash CP}$$

Optimal solutions to (Nash CP) coincide with Nash equilibria for the NCG with element latency functions $\vec{\ell}$. This has been noted by Beckmann et al. [11], Dafermos and Sparrow [37], and Braess [18, 19]. We will give a proof below.

A drawback in terms of practicability of (OPT NLP) and (Nash CP) is that they involve as many variables as strategies, which might be exponential in the number of elements. However, in the case of unicast routing we can remedy this by using flow-conservation rules as constraints, hence expressing everything on edge level. Paths do not occur explicitly anymore, which implies that for each commodity $i \in [N]$, *all* s_i-t_i paths are eligible for routing. If certain paths shall be excluded, this must be expressible by excluding certain edges, which may not always be the case.

Assume notation from Sec. 1.1 for the rest of this paragraph. For a vector $x = (x_e)_{e\in E}$ and a vertex $v \in V$ define

$$\partial(v, x) := \sum_{e\in\delta^{\text{out}}(v)} x_e - \sum_{e\in\delta^{\text{in}}(v)} x_e \ .$$

We need a vector $(x_e^{(i)})_{e\in E}$ for each source-terminal pair $i \in [N]$. Then

the constraints for unicast routing are:

$$\begin{aligned} x_e &= \sum_{i\in[N]} x_e^{(i)} && \forall\, e \in E \\ \partial(v, x^{(i)}) &= 0 && \forall\, v \in V \setminus \{s_i, t_i\}\,,\ \ i \in [N] \\ \partial(s_i, x^{(i)}) &= d_i && \forall\, i \in [N] \\ x_e^{(i)} &\geq 0 && \forall\, e \in E,\ \ i \in [N] \end{aligned}$$

These are only polynomially many (in N and m) constraints.

Now we prove that minimizing the potential function computes a Nash equilibrium.

1.12 Theorem. *A feasible $(x, a) \in \mathbb{R}^{m+n}_{\geq 0}$ is optimal for (Nash CP) if and only if the action distribution a is a Nash equilibrium.*

Proof. We give names to the constraints of (Nash CP): define $f_S(x, a) := -a_S$ for all $S \in \mathfrak{S}$, $h_i(x, a) := d_i - \sum_{S\in\mathfrak{S}_i} a_S$ for all $i \in [N]$, and $h_e(x, a) := x_e - \sum_{S\in\mathfrak{S}} r_{eS} a_S$ for all $e \in E$. Like the constraints, we will also consider Φ a function of $m + n$ variables, though it only depends on the first m. Slater's condition (see, e.g., [17, Sec. 5.2.3]) is easily verified: take $a_S := \frac{d_i}{n_i}$ for all $S \in \mathfrak{S}_i$ and $i \in [N]$, and $x_e := g_e(a)$ for all $e \in E$. Then (x, a) is in the interior of the domain of (Nash CP), i.e., the set where the objective function and all constraint functions are defined, which is $\mathbb{R}^n$ here. Moreover, $f_S(x, a) < 0$ for this point and all $S \in \mathfrak{S}$.

Hence a feasible (x, a) is optimal if and only if the Karush-Kuhn-Tucker (KKT) conditions hold. These are (see, e.g., [17, Sec. 5.5.3]): there exist real numbers $(\lambda_S)_{S\in\mathfrak{S}}, (\nu_i)_{i\in[N]}, (\nu_e)_{e\in E}$ such that $\lambda_S \geq 0$ and $\lambda_S a_S = 0$ for all $S \in \mathfrak{S}$ and

$$\nabla\Phi(x, a) + \sum_{S\in\mathfrak{S}} \lambda_S \nabla f_S(x, a) + \sum_{i\in[N]} \nu_i \nabla h_i(x, a) + \sum_{e\in E} \nu_e \nabla h_e(x, a) = 0\ .$$

This last equation is equivalent to

$$\begin{aligned} \ell_e(x_e) + \nu_e &= 0 && \forall e \in E \\ -\lambda_S - \nu_i - \sum_{e\in E} \nu_e r_{eS} &= 0 && \forall S \in \mathfrak{S}_i \quad \forall i \in [N]\ . \end{aligned}$$

It follows that a feasible (x, a) is optimal if and only if there exist $(\lambda_S)_{S\in\mathfrak{S}}, (\nu_i)_{i\in[N]}$ such that

$$\begin{aligned} \lambda_S \geq 0 \text{ and } \lambda_S a_S = 0 \quad & \forall S \in \mathfrak{S} \\ \sum_{e\in E} \ell_e(x_e) r_{eS} - \nu_i = \lambda_S \quad & \forall S \in \mathfrak{S}_i \quad \forall i \in [N] \ . \end{aligned}$$

We have $\sum_{e\in E} \ell_e(x_e) r_{eS} = L_S(a)$. Reformulating further, we see that a feasible (x, a) is optimal if and only if there exist $(\nu_i)_{i\in[N]}$ such that we have for all $i \in [N]$ and $S \in \mathfrak{S}_i$ the following: $L_S(a) - \nu_i \geq 0$, and $L_S(a) - \nu_i = 0$ in case of $a_S > 0$. This is the Nash condition (we have $\nu_i = \Lambda_i(a)$ if it holds). □

The implications of this result go beyond the computation of equilibria. Assume a standard class of element latency functions and for each $e \in E$ define the *marginal cost function* ℓ_e^* to be the derivative of $x \mapsto \ell_e(x)x$, i.e.,

$$\ell_e^* : \mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}_{\geq 0}, x \mapsto \ell_e'(x)x + \ell_e(x) \ .$$

Since ℓ_e is continuously differentiable, ℓ_e^* is continuous. Since $x \mapsto \ell_e(x)x$ is convex, the function ℓ_e^* is also non-decreasing,[2] and so it is eligible to be an element latency function in some NCG. For an NCG Γ let us denote Γ^* a modification of Γ where each ℓ_e is replaced by ℓ_e^*. Since $\int_0^x \ell_e^*(t)dt = \ell_e(x)x$, the potential for Γ^* coincides with SC for Γ.

1.13 Corollary. *An action distribution is a Nash equilibrium for* Γ^* *if and only if it is optimal for* Γ.

Proof. Follows from Thm. 1.12. □

This corollary is part of the motivation behind the techniques in the proof of Thm. 1.16 in the following section. More on this motivation is explained in detail (at the example of unicast routing) in [77, Sec. 3.2 and 3.3].

[2] This can be shown even if ℓ_e is not twice differentiable. For a real differentiable convex function f, we have $f(y) \geq f(x) + f'(x)(y - x)$. Hence $(f'(y) - f'(x))(y - x) \geq 0$ for all x, y.

1.7 Bounding the Price of Anarchy

We present three ways for bounding the price of anarchy in NCGs. For the first two, it is essential that we can express the social cost on strategy level as well as on element level, as shown in (1.2). Recall that responsible for this is that the rates of consumption appear in congestions as well as in strategy latencies. This equality also holds for the mixed social cost, i.e.,

$$\mathsf{SC}^a(b) = \sum_{S \in \mathfrak{S}} L_S(a) b_S = \sum_{e \in E} \ell_e(g_e(a)) g_e(b) \quad \text{for all } a, b \in \mathcal{A}. \tag{1.4}$$

1.7.1 The Anarchy Value

Let $\mathcal{L}$ be a standard class of element latency functions. For each $\ell \in \mathcal{L}$ choose a function $\mu_\ell : \mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}_{\geq 0}$ such that

$$\ell^*(\mu_\ell(v)) = \ell(v) \quad \text{and} \quad \mu_\ell(v) \leq v \quad \text{for all } v \in \mathbb{R}_{\geq 0}. \tag{1.5}$$

In the context of an NCG, we often write μ_e instead of μ_{ℓ_e}.

1.14 Proposition. *Let ℓ be a standard latency function. Then a function μ_ℓ, as given in (1.5), exists.*

Proof. The case $v = 0$ is clear by setting $\mu_\ell(0) := 0$. Fix $v > 0$, and define the function $\phi_v : \mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}, x \mapsto (\ell(v) - \ell(x))x$. Since $\phi_v(0) = \phi_v(v) = 0$, and, by monotonicity of ℓ, $\phi_v(x) \geq 0$ for all $x \leq v$, and $\phi_v(x) \leq 0$ for all $x \geq v$, it follows that ϕ_v attains its global maximum in some $x^* \in (0, v)$. Real calculus gives that $0 = \phi_v'(x^*) = \ell(v) - \ell'(x^*)x^* - \ell(x^*) = \ell(v) - \ell^*(x^*)$, hence $\ell^*(x^*) = \ell(v)$. Defining $\mu_\ell(v) := x^*$ concludes. □

Now we are ready to define the *anarchy value* [80, Def. 4.3] of $\mathcal{L}$, denoted $\alpha(\mathcal{L})$. Fix a choice of μ_ℓ, $\ell \in \mathcal{L}$; we will soon prove that the following definition is in fact independent of the actual choice of these

functions. Using the convention $\frac{0}{0} := 0$, we define

$$\begin{aligned}\alpha(\mathcal{L}) &:= \sup_{\substack{\ell\in\mathcal{L}\\ v\geq 0}} \frac{\ell(v)v}{(\ell(\mu_\ell(v)) - \ell(v))\,\mu_\ell(v) + \ell(v)v} \\ &= \sup_{\substack{\ell\in\mathcal{L}\\ v\geq 0}} \frac{\ell(v)v}{\ell(\mu_\ell(v))\,\mu_\ell(v) + \ell(v)\,(v - \mu_\ell(v))} \ .\end{aligned} \tag{1.6}$$

By the second formulation, we see that the denominator is always non-negative, since $\mu_\ell(v) \leq v$. If the denominator is 0 and $\ell(v) > 0$, then $v = \mu_\ell(v)$ and hence $\ell(v)v = \ell(\mu_\ell(v))\mu_\ell(v)$, which is 0 since the denominator is 0. So there is no division by 0 except $\frac{0}{0}$, which is defined to be 0.

Since $\mathcal{L}$ is standard, which means in particular that it contains a non-zero function, we moreover have $\alpha(\mathcal{L}) > 0$. The anarchy value does not need to be finite. However, we are only interested in cases where it is finite, and so we assume that in the following. For example, $\alpha(\mathrm{Poly}^+(p))$ is finite for any fixed $p \in \mathbb{N}_{\geq 1}$, as we will prove later.

Definition (1.6) implies

$$(\ell(\mu_\ell(v)) - \ell(v))\,\mu_\ell(v) + \ell(v)v \geq \frac{\ell(v)v}{\alpha(\mathcal{L})} \qquad \forall v \in \mathbb{R}_{\geq 0} \quad \forall \ell \in \mathcal{L} \ . \tag{1.7}$$

The following lemma will also be essential in the next section.

1.15 Lemma. *We have, using the convention $\frac{0}{0} := 0$ on the right-hand side:*

$$1 - \frac{1}{\alpha(\mathcal{L})} = \sup_{\substack{\ell\in\mathcal{L}\\ v\geq 0}} \frac{1}{\ell(v)v} \max_{x\geq 0} (\ell(v) - \ell(x))\,x \ . \tag{1.8}$$

Hence the anarchy value is independent of the actual choices for μ_ℓ, $\ell \in \mathcal{L}$.

Proof. The independence follows from (1.8), since the right-hand side is independent of any μ_ℓ, $\ell \in \mathcal{L}$. So we have to show (1.8), using an arbitrary choice of μ_ℓ, $\ell \in \mathcal{L}$ to define $\alpha(\mathcal{L})$. If for some $\ell \in \mathcal{L}$ and $v \geq 0$ the enumerator in definition (1.6) is 0, then by monotonicity of ℓ and definition of μ_ℓ the denominator is 0 as well. Using the convention

$\frac{0}{0} := 0$ we can hence reformulate the expression in (1.6) and receive:

$$1 - \frac{1}{\alpha(\mathcal{L})} = \sup_{\substack{\ell \in \mathcal{L} \\ v \geq 0}} \frac{1}{\ell(v)v} (\ell(v) - \ell(\mu_\ell(v)))\, \mu_\ell(v) \ .$$

Fix $\ell \in \mathcal{L}$ and $v > 0$ (the case $v = 0$ is clear) and consider again the function ϕ_v from the proof of Prop. 1.14, i.e., $\phi_v(x) = (\ell(v) - \ell(x))\, x$ for all $x \in \mathbb{R}_{\geq 0}$. Since $\mathcal{L}$ is standard, $x \mapsto \ell(x)x$ is convex, and so ϕ_v is concave. Hence *any* point $x^* \in (0, v)$ with $\phi_v'(x^*) = 0$ maximizes ϕ_v. The condition $\phi_v'(x^*) = 0$ is equivalent to $\ell^*(x^*) = \ell(v)$. So we have $\phi_v(\mu_\ell(v)) = \max_{x \geq 0} \phi_v(x)$ since $\ell^*(\mu_\ell(v)) = \ell(v)$. By definition of ϕ_v, it follows $(\ell(v) - \ell(\mu_\ell(v)))\, \mu_\ell(v) = \max_{x \geq 0} (\ell(v) - \ell(x))\, x$. □

Although the expression defining the anarchy value may look complicated at first, anarchy values for many interesting classes of functions can be derived relatively easily, see [77, Sec. 3.5.2]. For instance

$$\alpha(\mathrm{Poly}^+(p)) = (1 - p\,(p+1)^{-1-\frac{1}{p}})^{-1} = \Theta(\frac{p}{\ln p}) \ , \qquad (1.9)$$

exposing Pigou's example as a worst-case scenario. This also shows that the anarchy value may be ∞, e.g., if we consider the set of all polynomials with non-negative coefficients. We will give a formal derivation of (1.9) in Cor. 1.23.

Correa, Schulz, and Stier Moses [33] show that the concept of anarchy value can be simplified and generalized at the same time. This will be our topic in the next section; in fact we already provided the foundation for this in Lem. 1.15. Now we present a compact proof for the upper bound along the lines of Roughgarden and Tardos.

1.16 Theorem (Roughgarden and Tardos [80])**.** *Let $\mathcal{L}$ be a standard class of element latency functions. The price of anarchy of any NCG with element latency functions drawn from $\mathcal{L}$ is no more than $\alpha(\mathcal{L})$.*

Proof. For any three reals $x, y, z \in \mathbb{R}_{\geq 0}$ and an element $e \in E$ we have

$$\begin{aligned} \ell_e(y)y &= \ell_e(z)z + \int_z^y \ell_e^*(t)dt \geq \ell_e(z)z + \ell_e^*(z)\,(y - z) \\ &= (\ell_e(z) - \ell_e^*(z))\, z + \ell_e(x)x + \ell_e^*(z)y - \ell_e(x)x \ . \end{aligned}$$

We evaluate this for $z := \mu_e(x)$ and receive:

$$\begin{aligned}
&\ell_e(y)y \\
&\geq (\ell_e(\mu_e(x)) - \ell_e^*(\mu_e(x)))\,\mu_e(x) + \ell_e(x)x + \ell_e^*(\mu_e(x))y - \ell_e(x)x \\
&= (\ell_e(\mu_e(x)) - \ell_e(x))\,\mu_e(x) + \ell_e(x)x + \ell_e(x)y - \ell_e(x)x \quad \text{by def. } \mu_e \\
&\geq \frac{\ell_e(x)x}{\alpha(\mathcal{L})} + \ell_e(x)y - \ell_e(x)x \quad \text{by (1.7)}.
\end{aligned}$$

Let a be a Nash equilibrium and a^* an optimal action distribution. Evaluating the above for $x := g_e(a)$ and $y := g_e(a^*)$ and taking the sum over all $e \in E$ yields:

$$\begin{aligned}
&\mathsf{SC}(a^*) = \sum_{e\in E} \ell_e(g_e(a^*))g_e(a^*) \quad \text{by (1.4)} \\
&\geq \frac{1}{\alpha(\mathcal{L})} \sum_{e\in E} \ell_e(g_e(a))g_e(a) + \sum_{e\in E} \ell_e(g_e(a))g_e(a^*) - \sum_{e\in E} \ell_e(g_e(a))g_e(a) \\
&= \frac{1}{\alpha(\mathcal{L})}\mathsf{SC}(a) + \mathsf{SC}^a(a^*) - \mathsf{SC}(a) \overset{\text{by (VAR)}}{\geq} \frac{1}{\alpha(\mathcal{L})}\mathsf{SC}(a) \ .
\end{aligned}$$

This finishes the proof, since now we have $\frac{\mathsf{SC}(a)}{\mathsf{SC}(a^*)} \leq \alpha(\mathcal{L})$. □

The anarchy value bound is tight from a worst-case point of view, even when restricting to selfish unicast routing. We make this precise. A class $\mathcal{L}$ of latency functions is called *diverse*, if for each $y \in \mathbb{R}_{\geq 0}$ it contains a function ℓ such that $\ell(0) = y$. We cite the following theorem without proof.

1.17 Theorem (Roughgarden [77]). *Let $\varepsilon > 0$ and $\mathcal{L}$ be standard and diverse. Then there exists a unicast network consisting of one source, one terminal, and parallel edges in between with latency functions drawn from $\mathcal{L}$ such that its price of anarchy is at least $\alpha(\mathcal{L}) - \varepsilon$.*

Hence, *from a worst-case point of view*, the price of anarchy in unicast routing is independent of the network topology. On the other hand, for a particular instance, it may be dependent on the network topology; adding or removing edges may change the price of anarchy. A prominent example for this is Braess' Paradox [18, 19], see also Sec. 1.8, where an additional edge worsens the price of anarchy.

1.7.2 The β Parameter

Following Correa, Schulz, and Stier Moses [33], we present an extension of the concept of anarchy value, which acts as a simplification at the same time. It is also applicable for tightly bounding the price of anarchy in unicast routing where each edge in addition to its latency functions has got a *capacity*, which is the main theme of [33], but will not be treated here.

Let $\mathcal{L}$ be a class of continuous, non-decreasing functions $\mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}_{\geq 0}$, not necessarily standard. Using the convention $\frac{0}{0} := 0$, define

$$\beta(\mathcal{L}) := \sup_{\substack{\ell \in \mathcal{L} \\ v \geq 0}} \frac{1}{\ell(v)v} \max_{x \geq 0} (\ell(v) - \ell(x))\, x \ . \tag{1.10}$$

Monotonicity ensures that there is no division by zero, except $\frac{0}{0}$, which is defined to 0. It follows that

$$(\ell(v) - \ell(x))x \leq \beta(\mathcal{L})\, \ell(v)v \quad \forall v, x \in \mathbb{R}_{\geq 0} \quad \forall \ell \in \mathcal{L} \ . \tag{1.11}$$

It is easy to see that always $\beta(\mathcal{L}) \leq 1$. In case that $\mathcal{L}$ happens to be a standard class with finite anarchy value $\alpha(\mathcal{L})$, we have $\beta(\mathcal{L}) < 1$ and

$$\frac{1}{1 - \beta(\mathcal{L})} = \alpha(\mathcal{L}) \ . \tag{1.12}$$

This follows from Lem. 1.15.

A short proof for a bound on the price of anarchy including the result from Thm. 1.16 is possible using the parameter β.

1.18 Theorem (Correa, Schulz, Stier Moses [33])**.** *Let $\mathcal{L}$ be a class of continuous, non-decreasing functions $\mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}_{\geq 0}$ such that $\beta(\mathcal{L}) < 1$. The price of anarchy of any NCG with element latency functions drawn from $\mathcal{L}$ is no more than $\frac{1}{1-\beta(\mathcal{L})}$.*

Proof. Let a be a Nash equilibrium and a^* an optimal action distribution. Then

$$\mathsf{SC}(a) \leq \mathsf{SC}^a(a^*) \qquad \text{by (VAR)}$$

$$
\begin{aligned}
&= \sum_{e\in E} \big(\ell_e(g_e(a))g_e(a^*) - \ell_e(g_e(a^*))\, g_e(a^*)\big) + \mathsf{SC}(a^*) && \text{by (1.4)}\\
&= \sum_{e\in E} (\ell_e(\underbrace{g_e(a)}_{v:=}) - \ell_e(\underbrace{g_e(a^*)}_{x:=}))\ \underbrace{g_e(a^*)}_{x=} + \mathsf{SC}(a^*)\\
&\le \sum_{e\in E} \beta(\mathcal{L})\ell_e(g_e(a))g_e(a) + \mathsf{SC}(a^*) && \text{by (1.11)}\\
&= \beta(\mathcal{L})\mathsf{SC}(a) + \mathsf{SC}(a^*)\ .
\end{aligned}
$$

Hence $(1-\beta(\mathcal{L}))\,\mathsf{SC}(a) \le \mathsf{SC}(a^*)$, and so, using $\beta(\mathcal{L}) < 1$, we receive the claimed bound. □

Good upper bounds on β can be derived in a simple way and for large classes of functions, for instance for super- and sub-homogeneous classes. We will make use of this later in Sec. 2.9 and do all necessary preparations here.

1.19 Definition. Let $s : (0,1] \longrightarrow (0,\infty)$ such that $s(1) = 1$.

(i) An element latency function ℓ is called *s-super-homogeneous* if $\ell(\varepsilon x) \ge s(\varepsilon)\ell(x)$ for all $\varepsilon \in (0,1]$ and all $x \in \mathbb{R}_{\ge 0}$.

(ii) An element latency function class $\mathcal{L}$ is called *s-super-homogeneous* if each $\ell \in \mathcal{L}$ is s-super-homogeneous.

(iii) Define $\bar{s} : [1,\infty) \longrightarrow (0,\infty), t \mapsto s(t^{-1})^{-1}$.

1.20 Proposition. *An element latency function ℓ is s-super-homogeneous if and only if it is $\bar{s}$-*sub-homogeneous*, i.e.,*

$$\ell(tx) \le \bar{s}(t)\ell(x) \quad \forall t \ge 1 \quad \forall x \in \mathbb{R}_{\ge 0}\ . \tag{1.13}$$

Proof. If ℓ is s-super-homogeneous, we have

$$\ell(x) = \ell(t^{-1}tx) \ge s(t^{-1})\ell(tx) = \bar{s}(t)^{-1}\ell(tx)$$

for all $t \ge 1$ and all $x \in \mathbb{R}_{\ge 0}$, so ℓ is $\bar{s}$-sub-homogeneous. If, on the other hand, ℓ fulfills (1.13) then

$$\ell(x) = \ell(\varepsilon^{-1}\varepsilon x) \le \bar{s}(\varepsilon^{-1})\ell(\varepsilon x) = s(\varepsilon)^{-1}\ell(\varepsilon x)$$

for all $\varepsilon \in (0,1]$ and all $x \in \mathbb{R}_{\ge 0}$, so ℓ is s-super-homogeneous. □

1.21 Lemma ([33, Lem. 4.1]). *Let $\mathcal{L}$ be an s-super-homogeneous class of element latency functions. Then $\beta(\mathcal{L}) \leq \sup_{0<\varepsilon<1} \varepsilon(1-s(\varepsilon))$.*

Proof.

$$\begin{aligned}
\beta(\mathcal{L}) &= \sup_{\substack{\ell\in\mathcal{L}\\ 0\leq v}} \frac{1}{\ell(v)v} \max_{0\leq x} \left(\ell(v)-\ell(x)\right) x \\
&= \sup_{\substack{\ell\in\mathcal{L}\\ 0\leq v}} \frac{1}{\ell(v)v} \max_{0\leq x\leq v} \left(\ell(v)-\ell(x)\right) x && \text{by monotonicity} \\
&= \sup_{\substack{\ell\in\mathcal{L}\\ 0< v}} \frac{1}{\ell(v)v} \max_{0< x< v} \left(\ell(v)-\ell(x)\right) x \\
&= \sup_{\substack{\ell\in\mathcal{L}\\ 0< v}} \frac{1}{\ell(v)v} \max_{0< x< v} \left(\ell(v)-\ell(\tfrac{x}{v}v)\right) x \\
&\leq \sup_{\substack{\ell\in\mathcal{L}\\ 0< v}} \frac{1}{\ell(v)v} \sup_{0< x< v} \left(\ell(v)-s(\tfrac{x}{v})\ell(v)\right) x && \text{since } 0<\tfrac{x}{v}<1 \\
&= \sup_{0<\varepsilon<1} \varepsilon\left(1-s(\varepsilon)\right) \ .
\end{aligned}$$

□

A bound follows for $(\varepsilon \mapsto \varepsilon^p)$-super-homogeneous classes, such as $\mathrm{Poly}^+(p)$.

1.22 Corollary (cf. [33, Cor. 4.4]). *Let $\mathcal{L}$ be an $(\varepsilon \mapsto \varepsilon^p)$-super-homogeneous element latency function class. Then $\beta(\mathcal{L}) \leq p\,(p+1)^{-1-\frac{1}{p}}$.*

Proof.

$$\begin{aligned}
\beta(\mathcal{L}) &\leq \sup_{0<\varepsilon<1} \varepsilon - \varepsilon^{p+1} && \text{by Lem. 1.21} \\
&= (p+1)^{-\frac{1}{p}} - (p+1)^{-1-\frac{1}{p}} && \text{real calculus} \\
&= (p+1)\,(p+1)^{-1-\frac{1}{p}} - (p+1)^{-1-\frac{1}{p}} \\
&= p\,(p+1)^{-1-\frac{1}{p}} \ .
\end{aligned}$$

□

It follows an exact expression for $\beta(\mathrm{Poly}^+(p))$, and so $\alpha(\mathrm{Poly}^+(p))$, for some fixed $p \in \mathbb{N}_{\geq 1}$. Fig. 1.2 on the following page shows a plot.

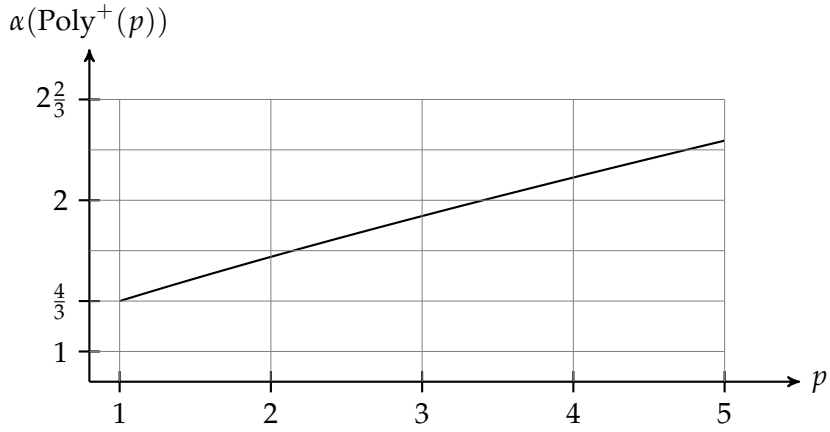


Figure 1.2. The anarchy value $\alpha(\mathrm{Poly}^+(p)) = \frac{1}{1-\beta(\mathrm{Poly}^+(p))}$ plotted for different values of p (including non-integer values). It is $\frac{4}{3}$ for $p = 1$ and ≈ 1.63 for $p = 2$.

1.23 Corollary. *We have* $\beta(\mathrm{Poly}^+(p)) = p(p+1)^{-1-\frac{1}{p}} < 1$ *and the asymptotic* $\alpha(\mathrm{Poly}^+(p)) = \frac{1}{1-\beta(\mathrm{Poly}^+(p))} = \Theta(\frac{p}{\ln p})$ *for* $p \to \infty$.

Proof. That $\beta(\mathrm{Poly}^+(p)) \leq p\,(p+1)^{-1-\frac{1}{p}}$ follows from Cor. 1.22. In particular

$$\beta(\mathrm{Poly}^+(p)) \leq p\,(p+1)^{-1-\frac{1}{p}} = \frac{p}{p+1}\,\frac{1}{(p+1)^{1/p}} < 1 \ .$$

That $\beta(\mathrm{Poly}^+(p)) \geq p(p+1)^{-1-\frac{1}{p}}$ follows from Pigou's example, see (1.1), and Thm. 1.18. The asymptotic can be shown by an invocation of l'Hôpital's rule. □

Parameter β is invariant under scaling of element latency functions:

1.24 Proposition. *Let* $\mathcal{L}$ *be a class of element lantency functions such that* $\beta(\mathcal{L}) < 1$. *Define*

$$\widetilde{\mathcal{L}} := \{x \mapsto c_1\ell(c_2x);\ c_1, c_2 \in \mathbb{R}_{>0}, \ell \in \mathcal{L}\} \ .$$

Then $\beta(\mathcal{L}) = \beta(\widetilde{\mathcal{L}})$.

Proof. Let $c_1, c_2 \in \mathbb{R}_{>0}$ and $\ell \in \mathcal{L}$. Then

$$\begin{aligned}
&\sup_{v \geq 0} \frac{1}{c_1 \ell(c_2 v) v} \max_{x \geq 0} \left(c_1 \ell(c_2 v) - c_1 \ell(c_2 x)\right) x \\
&= \sup_{v \geq 0} \frac{1}{\ell(c_2 v) v} \max_{x \geq 0} \left(\ell(c_2 v) - \ell(c_2 x)\right) x \\
&= \sup_{v \geq 0} \frac{1}{\ell(c_2 v) c_2 v} \max_{x \geq 0} \left(\ell(c_2 v) - \ell(c_2 x)\right) c_2 x \\
&= \sup_{v \geq 0} \frac{1}{\ell(v) v} \max_{x \geq 0} \left(\ell(v) - \ell(x)\right) x \ .
\end{aligned}$$

□

1.7.3 Jacobian Approach

This approach is due to Perakis [72] and works for non-atomic games as defined in Def. 1.4, under additional requirements. We only state the result for affine $\vec{L}$, say,

$$\vec{L} : \mathbb{R}^n \longrightarrow \mathbb{R}^n, x \mapsto Gx + b$$

for $G \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$. Then for the Jacobian of $\vec{L}$ we have $\mathrm{J}\vec{L}(x) = G$, independently of $x \in \mathbb{R}^n$. This matrix may be positive definite or not. If it is, and if $b^\top a \geq 0$ for all $a \in \mathcal{A}$, we can apply one of Perakis' bounds. So, let G be positive definite. Then $S \coloneqq (G + G^\top)/2$ is also positive definite (hence invertible) and symmetric. Let $\langle v, w \rangle_S \coloneqq v^\top S w$ for all $v, w \in \mathbb{R}^n$ and $\|v\|_S \coloneqq \sqrt{\langle v, v \rangle} = \sqrt{v^\top S v}$ for all $v \in \mathbb{R}^n$. Then $\langle \cdot, \cdot \rangle_S$ is a scalar product and $\|\cdot\|_S$ is a norm. Define

$$c(G) \coloneqq \|S^{-1} G\|_S = \sup_{\substack{v \in \mathbb{R}^n \\ v \neq 0}} \frac{\|S^{-1} G v\|_S}{\|v\|_S} \ ,$$

which is the operator norm of $S^{-1}G$ using $\|\cdot\|_S$ as the vector norm.

1.25 Theorem (Perakis [72]). *The price of anarchy in a non-atomic game with affine strategy latency function* $\vec{L} : \mathbb{R}^n \longrightarrow \mathbb{R}^n, x \mapsto Gx + b$ *with positive definite* G *and* b *such that* $b^\top a \geq 0$ *for all* $a \in \mathcal{A}$*, is bounded by*

$$\begin{cases} c^2 & \text{if } c^2 \geq 2 \\ \frac{4}{4-c^2} & \text{else} \end{cases},$$

where $c = c(G) = \|S^{-1}G\|_S$ *and* $S = (G + G^\top)/2$.

Proof. Let a be a Nash equilibrium and a^* an optimum. Then

$$\begin{aligned} \mathsf{SC}(a) &= \vec{L}(a)^\top a \\ &\leq \vec{L}(a)^\top a^* && \text{by (VAR)} \\ &= a^\top G^\top a^* + b^\top a^* \\ &= a^\top G^\top S^{-1} S\, a^* + b^\top a^* \\ &= (S^{-1}Ga)^\top S\, a^* + b^\top a^* && S^{-1} \text{ is symmetric} \\ &= \langle S^{-1}Ga, a^* \rangle_S + b^\top a^* \\ &\leq \|S^{-1}Ga\|_S \ \|a^*\|_S + b^\top a^* && \text{Cauchy-Schwarz} \\ &\leq \|a\|_S \ \|S^{-1}G\|_S \ \|a^*\|_S + b^\top a^* && \text{operator norm} \\ &= c \ \|a\|_S \ \|a^*\|_S + b^\top a^* && \text{definition.} \end{aligned}$$

We aim to turn the product $\|a\|_S \, \|a^*\|_S$ into a sum. For all $x, y \in \mathbb{R}_{\geq 0}$ and $t, u \in \mathbb{R}$ we have

$$0 \leq (\sqrt{x}\, t - \sqrt{y}\, u)^2 = xt^2 + yu^2 - 2\sqrt{xy}\, tu \ ,$$

hence

$$2\sqrt{xy}\, tu \leq xt^2 + yu^2 \ .$$

If we choose x, y such that $c \leq 2\sqrt{xy}$, and $x < 1 \leq y$, then this, using $t := \|a\|_S$ and $u := \|a^*\|_S$, implies

$$\begin{aligned} \mathsf{SC}(a) &\leq c \ \|a\|_S \ \|a^*\|_S + b^\top a^* \\ &\leq x\|a\|_S^2 + y\|a^*\|_S^2 + b^\top a^* \\ &= xa^\top Sa + ya^{*\top} Sa^* + b^\top a^* \end{aligned}$$

$$
\begin{aligned}
&= xa^\top Ga + ya^{*\top} Ga^* + b^\top a^* \\
&= x(a^\top Ga + b^\top a) + y(a^{*\top} Ga^* + b^\top a^*) - xb^\top a - (y-1)b^\top a^* \\
&\leq x\mathsf{SC}(a) + y\mathsf{SC}(a^*) \ .
\end{aligned}
$$

Using $x < 1$, it follows a bound on the price of anarchy

$$
\frac{\mathsf{SC}(a)}{\mathsf{SC}(a^*)} \leq \frac{y}{1-x} \ .
$$

We optimize:

$$
\begin{array}{lll}
\text{minimize} \quad \dfrac{y}{1-x} & \text{subject to} & c \leq 2\sqrt{xy} \quad (\iff \dfrac{c^2}{4} \leq xy) \\
& & 0 \leq x < 1 \\
& & 1 \leq y
\end{array}
$$

The objective function is increasing in x and y, so an optimal solution (x^*, y^*) has $\frac{c^2}{4} = x^*y^*$, i.e., $\frac{c^2}{4y^*} = x^*$. This leads to the equivalent formulation:

$$
\begin{array}{lll}
\text{minimize} \quad \dfrac{y^2}{y - \frac{c^2}{4}} & \text{subject to} & \dfrac{c^2}{4} < y \\
& & 1 \leq y
\end{array}
$$

The derivative of $y \mapsto y^2 \, (y - \frac{c^2}{4})^{-1}$ tells us that this function falls until $y = \frac{c^2}{2}$ and climbs after that.

- If $1 \leq \frac{c^2}{2}$ (i.e., $2 \leq c^2$), then we may choose $y := \frac{c^2}{2}$. It follows that the price of anarchy is bounded by $\frac{c^4/4}{c^2/4} = c^2$.
- If $\frac{c^2}{2} \leq 1$, we choose y as small as possible, i.e., $y = 1$. It follows that the price of anarchy is bounded by $\frac{1}{1-\frac{c^2}{4}} = \frac{4}{4-c^2}$. □

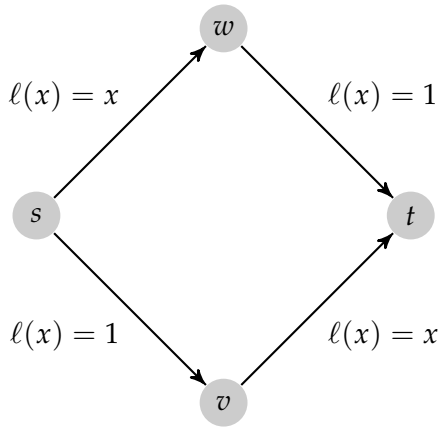
If G is only positive *semi*definite, Perakis proves another bound, which coincides with the former if G is positive definite. We refer to [72] for the details and also for the non-affine case.

1.8 Reasons for Inefficiency

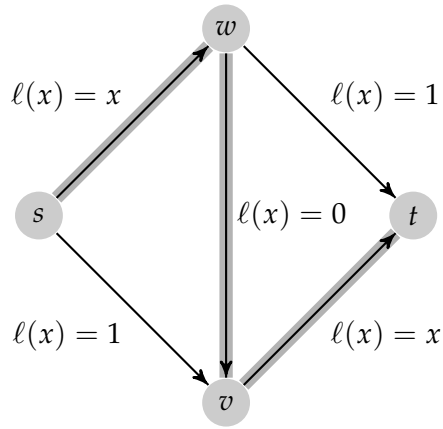
We point out two reasons for Nash equilibria being inefficient. Recall that in the preface, p. xiii, we named "absence of central authority" and "lack of cooperation" as the two principles under which we analyze the use or the forming of a network. Both can be recognized in the selfish unicast routing model. In Sec. 2.2 we will see how multicast routing can amplify the negative effects of lack of cooperation.

Absence of Central Authority. This is the actual selfishness. Unless forced to do so, no player would accept a higher latency than necessary, even if this impairs other players. Pigou's example shows what happens: in an optimal flow, half of the players enjoy a latency of $\frac{1}{2}$ on the lower link while the other half has to accept a latency of 1. In equilibrium, the lower link is so crowded that it has latency 1; everyone now experiences latency 1, there are no fast routes available anymore. It might appear that an optimum only demands that some players accept that other players are better off, but not that they have to accept a higher latency than in equilibrium. A slight modification of Pigou's example shows that this is not true in general. We reduce the demand to $1 - \varepsilon$ for an appropriate $\varepsilon > 0$. Then an optimal flow would still route a fraction of the demand on the upper link, forcing it to experience a latency of 1. In equilibrium, however, all demand flows on the lower link and experiences latency $1 - \varepsilon$, which is strictly smaller than 1. Roughgarden studies this "unfairness" of optima in detail [77, Sec. 2.4.5].

Lack of Cooperation. This can be seen by considering Braess' Paradox [18, 19]. Consider the instance shown in Fig. 1.3a on the next page with a demand of 1. An optimal as well as a Nash flow splits the demand evenly between the two alternative paths $P_1 := (s, w, t)$ and $P_2 := (s, v, t)$, resulting in a latency of $\frac{3}{2}$ for everyone. Now imagine that we build an additional link with latency function $\ell(x) = 0$ from w to v, as shown in Fig. 1.3b. This offers a third alternative path $P_3 := (s, w, v, t)$. The original flow f^* with $f^*_{P_1} = f^*_{P_2} = \frac{1}{2}$ remains optimal, but not a Nash equilibrium, for $L_{P_1}(f^*) = \frac{3}{2} > 1 = L_{P_3}(f^*)$.



a. Original network.



b. Including an additional link (w, v). In equilibrium, all flow travels on the highlighted path.

Figure 1.3. Braess' Paradox.

Instead, the flow f with $f_{P_3} = 1$, where all players take the new path, is a Nash equilibrium, since $L_{P_1}(f) = L_{P_2}(f) = L_{P_3}(f) = 2$. The latency experienced by everyone in equilibrium is worse, by factor $\frac{4}{3}$, than without the additional link. This may be unexpected, since intuition would suggest that additional links can only improve performance. It looks less paradoxical, however, when viewing the addition of a link as an extension of the set of available strategies and hence as giving players more 'opportunities' for – unfortunate – decisions. We refer to Roughgarden [77, Ch. 5] for an investigation into the algorithmic task of detecting such harmful links.

For our purposes in this section, the most important observation is that, given the network with the additional link, a lack of cooperation is the reason for the inefficiency of equilibrium. For, the flow f^*, splitting traffic across P_1 and P_2, is still an option in the new network. It would give everyone a better latency, it would even not lead to any player experiencing less latency than another one. Unfortunately, it is not a Nash equilibrium. However, if players would cooperate, they could agree on using f^*, i.e., sticking to P_1 or P_2, and not migrating to P_3. Such an agreement would be easy to reach since it would not require anyone making sacrifices.

1.9 Bibliographic Overview

We list some related work in roughly chronological order; this list is of course not complete. For comprehensive treatments of the price of anarchy we refer to the dissertation and the book by Roughgarden [77, 78] from 2002 and 2005 and the survey by Czumaj [35] from 2004.[3]

Pigou [73] in 1920 informally introduced what is nowadays known as "Pigou's Example". It is a simple unicast network exhibiting a high price of anarchy. See Sec. 1.1 for a formal treatment. We quote Pigou [73, p. 194]:

> Suppose there are two roads, ABD and ACD both leading from A to D. If left to itself, traffic would be so distributed that the trouble involved in driving a "representative" cart along each of the two roads would be equal. But, in some circumstances, it would be possible, by shifting a few carts from route B to route C, greatly to lessen the trouble of driving those still left on B, while only slightly increasing the trouble of driving along C.

Nash [69] in 1951 introduced the concept of non-cooperative games and their equilibria in an atomic model. The term "Nash equilibrium" was subsequently also used in other models to describe configurations in which no player has an incentive to deviate from the chosen strategy, given the strategies of the other players fixed. Wardrop [87] in 1952 studied a non-atomic model in the context of road traffic. He introduced two principles, which are the base of our understanding of Nash equilibria and optimal action distributions in the non-atomic case, respectively. We quote Wardrop [87, p. 345], cf. p. 12:

> (1) The journey times on all the routes actually used are equal, and less than those that would be experienced by a single vehicle on any unused route.
>
> (2) The average journey time is a minimum.

The characterization of Nash equilibria as optimal solutions to a convex program was noted by Beckmann, McGuire, and Winsten [11] in 1956 in the context of road traffic, Dafermos and Sparrow [37] in

[3] However, those clearly cannot cover some of the more recent works cited below.

1969, and Braess [18, 19] in 1968. Smith [84] in 1979 studied existence and uniqueness issues in the Wardrop model and gave the variational inequality formulation in Thm. 1.6, which is frequently used in this thesis.

Rosenthal [76] in 1973 introduced congestion games in an atomic setting. Existence of equilibria was proved by use of a potential function. Schmeidler [83] in 1973 introduced a general class of non-atomic games and gave existence results for equilibria; see also the notes by Mas-Colell [60] from 1984 and Rath [75] from 1992.

In 1999, Koutsoupias and Papadimitriou [54] in an atomic setting of routing in a simple network studied the quantity that would soon later be named "price of anarchy" [71]. Their model, also known as the "KP model", and variations of it were subsequently addressed, among others, by Czumaj and Vöcking [36] in 2002, Lücking, Mavronicolas, Monien, Rode, Spirakis, and Vrto [58] in 2003, and Lücking, Mavronicolas, Monien, Rode [59] in 2004. The price of anarchy in atomic versions of selfish unicast routing or congestion games has been studied by Fotakis, Kontogiannis, and Spirakis [42] in 2004, Christodoulou and Koutsoupias [28], Awerbuch, Azar, and Epstein [6] and Roughgarden [79] in 2005, Cominetti, Correa, and Stier Moses [30] in 2006, and Yang, Han, and Lo [90] in 2008.

On the non-atomic side, the dissertation of Roughgarden [77] in 2002 and the work of Roughgarden and Tardos [80] introduced the anarchy value, which has shown to be a strong concept for bounding the price of anarchy; we presented part of their work in Sec. 1.7.1. Subsequently, in 2004, that concept was generalized and simplified by Correa, Schulz, and Stier Moses [33]; we presented part of their work, around the β parameter, in Sec. 1.7.2. Recent work on atomic models, e.g. [30], uses a similar parameter. Correa et al. recently gave a graphical proof [34] for a bound on the price of anarchy. Chau and Sim [25] in 2003 and Perakis [72] in 2004 and Han, Lo, and Yang [47] in 2008 studied the price of anarchy in generalized settings; we presented part of Perakis' work in Sec. 1.7.3. Milchtaich [63] in 2004 studied efficiency of equilibria and cooperation aspects in a non-atomic model related to that of [83], with the restriction "that each player's payoff is only affected by the measure of the set of players whose choice of action

or facility is the same as his". In 2005 and 2006 [64, 65], he studied the effect of network topology on uniqueness and efficiency of equilibria in settings related to selfish unicast routing.

More results on non-atomic models were published by Friedman [43] in 1996, Milchtaich in a series of works beginning in 1996, e.g., [61, 62] and those cited above, and Blonski [15] in 1999.

Chapter 2

Non-Atomic Consumption-Relevance Congestion Games (NCRCG)

We present the main *new* results in this chapter on selfish multicast routing and the encompassing NCRCG model. The starting point is an extension of selfish unicast routing to selfish multicast routing. In Sec. 2.1 we show how selfish multicast routing can be modeled. In Sec. 2.2 we give some examples, proving non-uniqueness of Nash equilibrium and a lower bound on the price of anarchy – both results exposing new phenomena not existing in the unicast case. In Sec. 2.3 we explain the relation to NCGs, showing that some, but not all, variants of selfish multicast routing can be modeled by NCGs. Motivated by the limitations of the NCG model, we introduce the NCRCG model in Sec. 2.4 and Sec. 2.5. The difference compared to the NCG model is that we have two sets of rates of consumption, called *consumption numbers* and *relevance numbers*, respectively. In Sec. 2.6 we show how NCRCGs can be used to model more variants of selfish multicast routing than NCGs can do. In Sec. 2.7 to Sec. 2.10 we study the price of anarchy in the NCRCG model in depth. The most important result probably is the upper bound on the price of anarchy in Thm. 2.12; it depends on the β parameter and our newly introduced γ parameter (Sec. 2.5). The known bounds on NCGs follow as a corollary for the special case $\gamma = 1$. In Sec. 2.11 we consider computational issues. In Sec. 2.12 a few

bibliographic remarks are made. We give a summary of our results in Sec. 2.13.

2.1 Selfish Multicast Routing

Again, we have a directed multigraph (V, E) modeling a communication network, where each edge $e \in E$ has a latency function ℓ_e. We are given N sources s_i, $i \in [N]$, each of them associated with $k_i \in \mathbb{N}_{\geq 1}$ distinct terminals $t_i^1, \ldots, t_i^{k_i}$. We call $(s_i, \{t_i^1, \ldots, t_i^{k_i}\})$ a *commodity*. To avoid notational conflicts, we assume that all commodities are distinct. For each $i \in [N]$, a demand of d_i has to be routed from s_i to each of the terminals $t_i^1, \ldots, t_i^{k_i}$ simultaneously. Again, we assume the demand to represent a large number of selfish players, each controlling only a negligible fraction of it.

To realize the desired routing for some fraction of the demand d_i for some i, we have to choose k_i paths $S := \{P_1, \ldots, P_{k_i}\}$, where P_j connects s_i with t_i^j for each $j \in [k_i]$. Such a set S will take the role that a single path has in unicast routing. Anticipatory, we adopt terminology from non-atomic games and call such a set of paths a *strategy*. Let $\mathfrak{S}_i$ be the set of all eligible strategies for i and $\mathfrak{S} := \dot{\bigcup}_{i \in [N]} \mathfrak{S}_i$. For each $S \in \mathfrak{S}$ and $e \in E$ denote $S(e) := \{P \in S;\ e \in E(P)\}$, i.e., those paths in S containing e. Denote also $E(S) := \bigcup_{P \in S} E(P)$.

A *flow* is a vector $f : \mathfrak{S} \longrightarrow \mathbb{R}_{\geq 0}$ such that $\sum_{S \in \mathfrak{S}_i} f_S = d_i$ for all $i \in [N]$. So far, everything has been roughly analogous to unicast; we merely switched from paths to sets of path as strategies. The main differences start now. There are at least two ways to *realize* a flow using a strategy S. One is to realize a flow in the usual sense, obeying the Kirchhoff flow conservation rule: the amount of flow entering a node is exactly the amount of flow leaving the node (except sources and terminals). This is called a *conservation flow*. A smarter way to realize a flow using S is to exploit that we deal with *data* to be routed. Unlike physical flows, data can be duplicated virtually without cost, provided that the nodes in the network offer such a feature. Thus the same data has to be sent down an edge only *once*, no matter how many terminals are served by this. We call this *duplication flows*.

Whether we use conservation or duplication flows, decides how congestions look like. Clearly, duplication flows tend to cause fewer congestion. We define for each edge $e \in E$ and flow f:

$$g_e(f) \coloneqq \sum_{S \in \mathfrak{S}} |S(e)|\, f_S \qquad \text{for conservation flows,}$$

$$g_e(f) \coloneqq \sum_{\substack{S \in \mathfrak{S}: \\ S(e) \neq \emptyset}} f_S \qquad \text{for duplication flows.}$$

So for conservation flows we sum over all strategies S, and for each such S we determine how many paths it contains that in turn contain the edge e. This is the number $|S(e)|$; it is 0 if S does not use e at all. In other words, this number gives how many terminals are served through e when using S. To determine the congestion of e, we have to multiply f_S with this number. For, f_S gives a fraction of the demand that is routed to all the terminals simultaneously.

For duplication flows, we only have to determine for a given S whether some path in it contains edge e. If so, then S contributes f_S to the congestion. For, it does not matter how many terminals are served here, the data has just to be sent once.

Now we turn to strategy latencies. There are many different reasonable ways in which the latency of a strategy S, i.e., the latency of a set of paths, can be understood. Four of them were pointed out by Baltz, Esquivel, Kliemann, and Srivastav in [9]. We focus on two of them here, namely:

$$L_S(f) \coloneqq \sum_{e \in E(S)} \ell_e(g_e(f)) \qquad \textit{edge-based} \text{ latency}$$

$$L_S(f) \coloneqq \sum_{e \in E} |S(e)|\, \ell_e(g_e(f)) \qquad \textit{path-based} \text{ latency}$$

The latter can also be expressed via latencies of paths. Let the latency of a path P, denoted $L_P(f)$, be as in unicast routing, i.e., $L_P(f) \coloneqq \sum_{e \in E(P)} \ell_e(g_e(f))$. Then we have

$$L_S(f) = \sum_{P \in S} L_P(f) \qquad \text{for path-based latency.}$$

For a motivation of path-based latency, think of the source as a service provider and the terminals as customers which pay according to the quality of service. A higher latency means a loss of income for the service provider, which is proportional to the number of customers that experience that higher latency. The provider will hence consider an edge more important if it serves many customers.

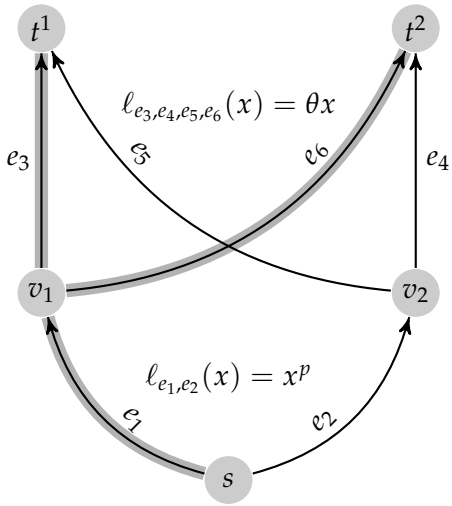
2.2 Non-Uniqueness of Equilibrium and Lower Bound

We give an example instance for selfish multicast routing using the combination of *duplication flows* and *path-based latency*. The example serves two purposes: we show non-uniqueness of Nash equilibrium, which is a novelty compared to unicast (cf. Sec. 1.5), and we show a lower bound on the price of anarchy that breaks the anarchy value of the involved edge latency functions (cf. Sec. 1.7.1). Consider the multicast instance from Fig. 2.1 on the facing page. We have one source, two terminals, and four strategies $S_1, \dots, S_4$ available (these are all strategies that this network offers). Each strategy consists of two paths. For example, strategy S_1 consists of paths (s, e_1, v_1, e_3, t^1) to reach terminal t^1 and (s, e_1, v_1, e_6, t^2) to reach terminal t^2. Edges e_1 and e_2 have element latency function $x \mapsto x^p$, and edges e_3 to e_6 have element latency function $x \mapsto \theta x$, where $p \in \mathbb{N}_{\geq 1}$ and $\theta \in \mathbb{R}_{\geq 0}$ are parameters.

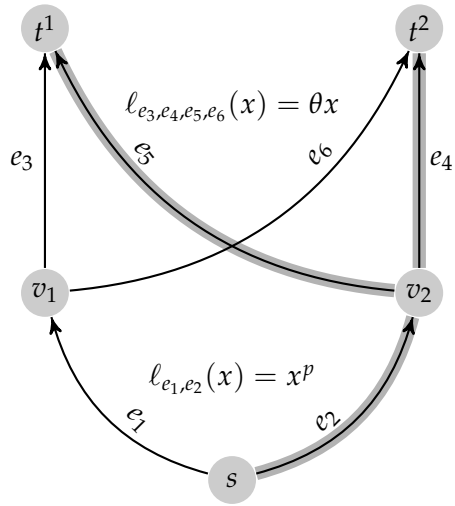
We write out strategy latencies for all four strategies for an arbitrary flow f:

$$\begin{aligned}
L_{S_1}(f) &= 2\ell_{e_1}(g_{e_1}(f)) + \ell_{e_3}(g_{e_3}(f)) + \ell_{e_6}(g_{e_6}(f)) \\
&= 2g_{e_1}(f)^p + \theta g_{e_3}(f) + \theta g_{e_6}(f) \\
L_{S_2}(f) &= 2\ell_{e_2}(g_{e_2}(f)) + \ell_{e_5}(g_{e_5}(f)) + \ell_{e_4}(g_{e_4}(f)) \\
&= 2g_{e_2}(f)^p + \theta g_{e_5}(f) + \theta g_{e_4}(f) \\
L_{S_3}(f) &= \ell_{e_1}(g_{e_1}(f)) + \ell_{e_3}(g_{e_3}(f)) + \ell_{e_2}(g_{e_2}(f)) + \ell_{e_4}(g_{e_4}(f)) \\
&= g_{e_1}(f)^p + g_{e_2}(f)^p + \theta g_{e_3}(f) + \theta g_{e_4}(f) \\
L_{S_4}(f) &= \ell_{e_1}(g_{e_1}(f)) + \ell_{e_6}(g_{e_6}(f)) + \ell_{e_2}(g_{e_2}(f)) + \ell_{e_5}(g_{e_5}(f)) \\
&= g_{e_1}(f)^p + g_{e_2}(f)^p + \theta g_{e_5}(f) + \theta g_{e_6}(f)
\end{aligned}$$

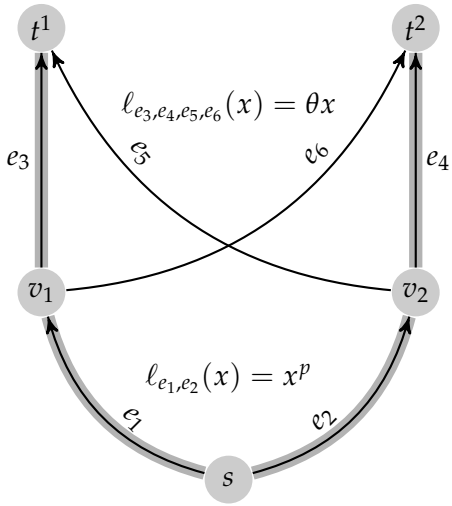
Let the demand be $d = 1$. It follows that $f^* := (\frac{1}{2}, \frac{1}{2}, 0, 0)$ and $f' := (0, 0, \frac{1}{2}, \frac{1}{2})$



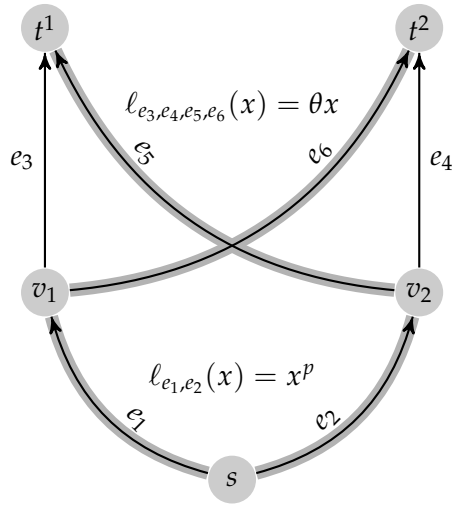
$S_1 := \{(s, e_1, v_1, e_3, t^1), (s, e_1, v_1, e_6, t^2)\}$



$S_2 := \{(s, e_2, v_2, e_5, t^1), (s, e_2, v_2, e_4, t^2)\}$



$S_3 := \{(s, e_1, v_1, e_3, t^1), (s, e_2, v_2, e_4, t^2)\}$



$S_4 := \{(s, e_2, v_2, e_5, t^1), (s, e_1, v_1, e_6, t^2)\}$

Figure 2.1. Multicast instance admitting multiple Nash equilibria with different congestions and different social cost. Parameters are $p \in \mathbb{N}_{\geq 1}$ and $\theta \in \mathbb{R}_{\geq 0}$. The instance is shown four times to highlight all four strategies $S_1, \ldots, S_4$.

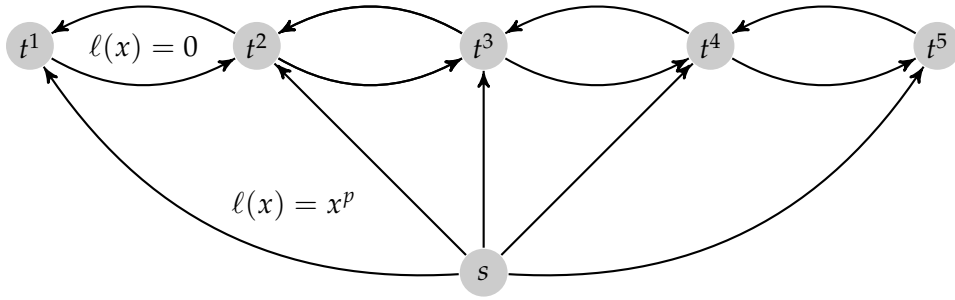


Figure 2.2. Example from [9] with $k = 5$ terminals. The links between the terminals have latency function $\ell(x) = 0$. The links from the source s to each terminal have latency function $\ell(x) = x^p$.

are both Nash equilibria, since they induce the following congestions:

$$g_e(f^*) = \frac{1}{2} \quad \forall e \in \{e_1, \ldots, e_6\} \ ,$$
$$g_e(f') = 1 \quad \forall e \in \{e_1, e_2\} \quad \text{and} \quad g_e(f') = \frac{1}{2} \quad \forall e \in \{e_3, e_4, e_5, e_6\} \ ,$$

and so $L_S(f^*) = (\frac{1}{2})^{p-1} + \theta$ and $L_S(f') = 2 + \theta$ for all $S \in \{S_1, \ldots S_4\}$. These two flows have different social cost, namely $\mathsf{SC}(f^*) = (\frac{1}{2})^{p-1} + \theta$ and $\mathsf{SC}(f') = 2 + \theta$, and also induce different element latencies. This is an example that in this form of selfish multicast routing, there may be Nash equilibria of different social cost and different congestions, even though all element latency functions are strictly increasing (consider case $\theta > 0$).

Regarding the price of anarchy, we note that it is at least

$$\frac{\mathsf{SC}(f')}{\mathsf{SC}(f^*)} = 2^{p-1} \frac{2 + \theta}{1 + \theta 2^{p-1}} = \Omega(2^p) \quad \text{for } \theta = O(2^{1-p}).$$

This not only breaks the anarchy value bound, which would have been $O(\frac{p}{\ln p})$, but also shows a huge gap between the social costs of different Nash equilibria. A similar example can be constructed with k terminals, for any number $k \in \mathbb{N}$, yielding a lower bound of $\Omega(k^p)$, as shown in [9]. The construction is given in Fig. 2.2. Latency functions

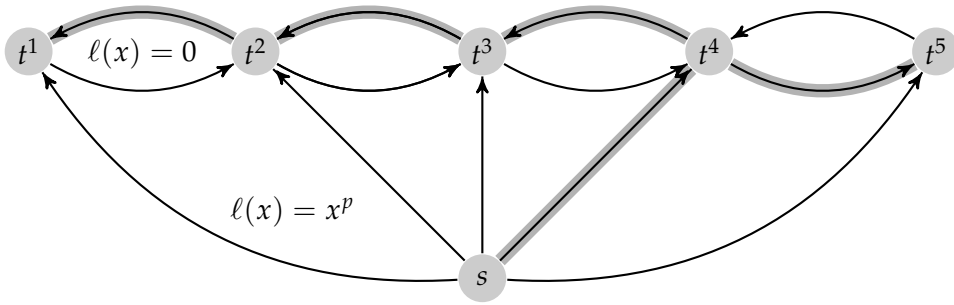


Figure 2.3. Example from Fig. 2.2 with all links participating in any path from strategy S_4 highlighted.

are $\ell_{(s,t^i)}(x) = x^p$ for all $i \in \{1,\dots,k\}$, $\ell_{(t^i,t^{i+1})}(x) = 0$ for all $i \in \{1,\dots,k-1\}$, and $\ell_{(t^i,t^{i-1})}(x) = 0$ for all $i \in \{2,\dots,k\}$. In other words, the source s has a connection with latency $x \mapsto x^p$ to each terminal, and neighboring terminals are connected in both directions with very fast links, having latency 0. Let the demand be $d = 1$. Define strategies, i.e., sets of paths $S_0,\dots,S_k$, by

$$\begin{aligned} S_0 &:= \{(s,t^i);\ i \in [k]\} \\ S_i &:= \{(s,t^i,t^{i+1},\dots,t^{i+j});\ j \in \{0,\dots,k-i\}\} \\ &\quad \cup \{(s,t^i,t^{i-1},\dots,t^{i-j});\ j \in \{0,\dots,i-1\}\} \qquad \text{for } i \in [k]. \end{aligned}$$

S_0 means sending traffic directly to each terminal, not making use of duplication. S_i means 'injecting' traffic along the link (s,t^i) into the row of terminals and then propagating it via the fast links to all terminals, using duplication. There are more strategies than this available, but we will only need these to define our flows. Fig. 2.3 shows an illustration of one such a strategy.

Define f by $f_{S_0} := 1$ and $f_{S_i} := 0$ for all $i \in [k]$. Then $\mathsf{SC}(f) = k$ and f is a Nash equilibrium. This can be seen as follows. All links emanating from the source s have latency 1 under f. So each path in any strategy has latency at least 1, and since there are k paths in each strategy, any strategy has latency at least k (in fact exactly k). Since $L_{S_0}(f) = k$, we have a Nash equilibrium.

For comparison, put $f^*_{S_0} := 0$ and $f^*_{S_i} := \frac{1}{k}$ for each $i \in [k]$. Then we have strategy latency for each $i \in [k]$:

$$\begin{aligned} L_{S_i}(f^*) &= k\,\ell_{(s,t^i)}\big(g_{(s,t^i)}(f^*)\big) && \text{link } (s,t^i) \text{ serves } k \text{ terminals} \\ &= k\,\ell_{(s,t^i)}\big(\frac{1}{k}\big) && \text{definition } f^* \text{ and congestion} \\ &= k\,\frac{1}{k^p} = k^{1-p} && \text{definition link latency function.} \end{aligned}$$

It follows for the social cost:

$$\begin{aligned} \mathsf{SC}(f^*) &= \sum_{i=0}^{k} L_{S_i}(f^*) f^*_{S_i} && \text{definition social cost} \\ &= \sum_{i=1}^{k} L_{S_i}(f^*) f^*_{S_i} && \text{definition } f^*_{S_0} \\ &= \sum_{i=1}^{k} L_{S_i}(f^*) \frac{1}{k} && \text{definition } f^*_{S_1}, \ldots, f^*_{S_k} \\ &= \sum_{i=1}^{k} k^{1-p} \frac{1}{k} = k^{1-p} && \text{previous calculation.} \end{aligned}$$

It follows that the price of anarchy is at least $\frac{\mathsf{SC}(f)}{\mathsf{SC}(f^*)} = \frac{k}{k^{1-p}} = k^p$.

Lack of Cooperation. These examples work by exploiting lack of cooperation. In the inefficient flows, players block out the links that emanate directly from the source, making all possible strategies slow for everyone. They do not even make use of duplication; each strategy connects the source with each of the terminals via pairwise edge-disjoint paths. With cooperation, they could agree on using f^*, which would be beneficial for everyone.

We make a general observation concerning the relation of price of stability and lack of cooperation. Recall Prop. 1.5. The following is restricted to the case that we have only one player class. It follows from that proposition that a more efficient equilibrium (i.e., one with lower social cost) provides lower latencies for everyone. Therefore, a gap between price of stability and price of anarchy originates only from a

lack of cooperation. In other words, a sufficient degree of cooperation could reduce the price of anarchy to the price of stability. This holds for general non-atomic games (as per Def. 1.4) with one player class.

2.3 Modeling as NCGs

The lower bound on the price of anarchy shown in the previous section already proves that there is no way to model selfish multicast routing with duplication flows and path-based latency as an NCG by using the edge latency functions as element latency functions. Such a modeling would imply a price of anarchy bounded by the anarchy value, which is impossible as we have just seen. However, two other variants can be modeled as NCGs:

2.1 Theorem. *Selfish multicast routing in the following settings can be modeled as an NCG that uses the edge latency functions as element latency functions:*

- *Conservation flows combined with path-based latency.*
- *Duplication flows combined with edge-based latency.*

Proof. We let network links correspond to elements, sets of paths eligible for routing (which we already called "strategies") to strategies, and source-terminal pairs to player classes. Recall the matrix notation from Sec. 1.3. Abusing notation, we use network links and strategies (i.e., sets of paths) as indices for the matrix rows and columns, respectively. All we are now left to do with is to define the correct rates of consumption.

- For conservation flows combined with path-based latency, for each $e \in E$ and $S \in \mathfrak{S}$ we define $r_{eS} := |S(e)|$.
- For duplication flows combined with edge-based latency, for each $e \in E$ and $S \in \mathfrak{S}$ we define

$$r_{eS} := \begin{cases} 1 & \text{if } e \in E(S) \\ 0 & \text{else} \end{cases} . \tag{2.1}$$

It is easily verified that the resulting congestions and strategy latencies

in the NCG coincide with the respective notions of selfish multicast routing. □

It follows with Thm. 1.18:

2.2 Corollary. *Let $\mathcal{L}$ be a class of element latency functions with $\beta(\mathcal{L}) < 1$. The price of anarchy in the selfish multicast routing variants listed in the previous theorem using edge latency functions from $\mathcal{L}$ is upper-bounded by $\frac{1}{1-\beta(\mathcal{L})}$.*

The proof of Thm. 2.1 gives insight why we cannot model the combination of duplication flows and path-based strategy latency as an NCG in that way. If we were to follow the same approach, then on the one hand, we would need r_{eS} to be as in (2.1) in order to have the correct congestions. On the other hand, however, we would need $r_{eS} = |S(e)|$ in order to have the correct strategy latencies. The same observation in reverse holds for the combination of conservation flows and edge-based latency.

It appears that we need *two different sets of rates of consumption* – one for congestions and one for strategy latencies. This generalization is accomplished by the model of *non-atomic consumption-relevance congestion games*, shortly "NCRCG". We will introduce these games in the next section and subsequently study their price of anarchy and computational issues.

2.4 Non-Atomic Consumption-Relevance Congestion Games

Recall the matrix notation for NCGs from Sec. 1.3. The new model is essentially the NCG model with *two* matrices: one for congestions and one for strategy latencies. We give a self-contained definition.

2.3 Definition. An instance in the non-atomic consumption-relevance congestion game model, shortly referred to as "an NCRCG", is defined by the following four items:

(i) Two numbers $m, n \in \mathbb{N}_{\geq 1}$. Denote $E := [m]$ and $\mathfrak{S} := [n]$, and call them *elements* and *strategies*, respectively.

(ii) For each $e \in E$ a real, continuous, non-decreasing function $\ell_e : \mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}_{\geq 0}$, called the *element latency function* of e.

(iii) Two matrices $C, R \in \mathbb{R}_{\geq 0}^{m \times n}$ such that for all $e \in E$ and $S \in \mathfrak{S}$ we have

$$C_{eS} = 0 \text{ if and only if } R_{eS} = 0 \ . \tag{2.2}$$

We call entries in C *consumption numbers* and entries in R *relevance numbers*. We require that there is at least one non-zero entry in each column.[1]

(iv) Numbers $d_1, \ldots, d_N \in \mathbb{R}_{>0}$, called *demands*, and a partition $\mathfrak{S} = \bigcup_{i \in [N]} \mathfrak{S}_i$. We call each $i \in [N]$ a *player class*; so each player class i has a demand d_i and some strategies $\mathfrak{S}_i$.

All the following notions are relative to a fixed NCRCG. An *action distribution* is a vector $a \in \mathbb{R}_{\geq 0}^n$ such that $\sum_{S \in \mathfrak{S}_i} a_S = d_i$ for all $i \in [N]$. Let $\mathcal{A}$ be the set of all action distributions. We define *congestion*, *strategy latency*, and *social cost*, respectively:

$$\begin{aligned} g_e(a) &:= \sum_{S \in \mathfrak{S}} C_{eS} a_S && \text{for } a \in \mathcal{A} \text{ and } e \in E \\ L_S(a) &:= \sum_{e \in E} R_{eS} \ell_e(g_e(a)) && \text{for } a \in \mathcal{A} \text{ and } S \in \mathfrak{S} \\ \mathsf{SC}(a) &:= \sum_{S \in \mathfrak{S}} L_S(a) a_S && \text{for } a \in \mathcal{A} \end{aligned}$$

For an action distribution $a \in \mathcal{A}$ and element $e \in E$ we call $\ell_e(g_e(a))$ the *element latency* of e under a.

Denote also for an action distribution $a \in \mathcal{A}$ the following vectors:

$$\begin{aligned} \vec{g}(a) &:= (g_e(a))_{e \in E} \in \mathbb{R}_{\geq 0}^m \\ \vec{\ell}(a) &:= (\ell_e(g_e(a))_{e \in E} \in \mathbb{R}_{\geq 0}^m \\ \vec{L}(a) &:= (L_S(a))_{S \in \mathfrak{S}} \in \mathbb{R}_{\geq 0}^n \end{aligned}$$

Then the social cost can be written as a scalar product $\mathsf{SC}(a) = \vec{L}(a)^\top a$. The vector of congestions can also be written as a matrix-vector product $\vec{g}(a) = Ca$.

[1] We know this requirement already from the previous chapter. The same justification holds here, cf. p. 10. We can also anytime require without loss of generality that there are no *rows* of only zeros, or in other words, that each element is contained in at least one strategy.

The congestion function $\vec{g}(\cdot)$ is in fact, by a straightforward extension, defined for all vectors from $\mathbb{R}^n_{\geq 0}$ and not only for action distributions. Since element latency functions are defined on $\mathbb{R}_{\geq 0}$, also functions $\vec{\ell}$, $\vec{L}$, and SC are defined for vectors from $\mathbb{R}^n_{\geq 0}$. We will make use of that in a few places; in particular, we will use it later when scaling the demands and speaking of action distributions with respect to those new demands.

2.4 Definition.

(i) An action distribution $a \in \mathcal{A}$ is called a *Nash equilibrium*, abbreviated "N.E.", if

$$\forall i \in [N] \quad \forall S, T \in \mathfrak{S}_i : \quad \big(a_S > 0 \implies L_S(a) \leq L_T(a) \big) \ .$$

(ii) An action distribution $a^* \in \mathcal{A}$ is called *optimal*, or an *optimum*, if

$$\mathsf{SC}(a^*) = \min_{a \in \mathcal{A}} \mathsf{SC}(a) \ .$$

Denote $\mathsf{OPT} \coloneqq \mathsf{SC}(a^*)$ for an optimum a^*.

An NCRCG is a non-atomic game with continuous strategy latency functions. Therefore, by the results in Sec. 1.4.2, Nash equilibria exist and are characterized by the variational inequality (VAR) on p. 14. By continuity of $\vec{L}$ and compactness of $\mathcal{A}$, optima exist as well, and we maintain our assumption of $\mathsf{OPT} > 0$.

2.5 Definition. We define the *price of stability* and the *price of anarchy* of an NCRCG, respectively, by

$$\inf_{\substack{a \in \mathcal{A} \\ a \text{ is N.E.}}} \frac{\mathsf{SC}(a)}{\mathsf{OPT}} \qquad \text{and} \qquad \sup_{\substack{a \in \mathcal{A} \\ a \text{ is N.E.}}} \frac{\mathsf{SC}(a)}{\mathsf{OPT}} \ .$$

By continuity, we can replace the infimum and supremum by minimum and maximum, respectively.

If $C = R$, then the NCRCG model coincides with the NCG model. Hence, all results that hold for NCGs also hold for NCRCGs with $C = R$. The general case $C \neq R$, however, can behave very differently from NCGs, as we have already seen. The rest of this chapter deals

with the price of anarchy in the general case and with the computation of optima and Nash equilibria.

2.5 New Parameters and Global Measures

Let an NCRCG be given. Using the convention $\frac{0}{0} := 0$, we introduce three new parameters:

$$\gamma_1 := \max_{e\in E, S\in\mathfrak{S}} \frac{C_{eS}}{R_{eS}} \ , \qquad \gamma_2 := \max_{e\in E, S\in\mathfrak{S}} \frac{R_{eS}}{C_{eS}} \ , \qquad \gamma := \gamma_1\gamma_2 \ .$$

By (2.2), no division by 0 occurs, except $\frac{0}{0} = 0$. Since a maximum is taken, it follows that 0-entries in C and R do not influence these parameters.

2.6 Proposition.

(i) It is $\gamma_1, \gamma_2 > 0$.

(ii) It is $\gamma \geq 1$.

(iii) If $C = R$, *i.e., in case of an NCG, it is* $\gamma_1 = \gamma_2 = \gamma = 1$.

Proof. (i) Every column of C and R has at least one row with a positive entry in both C and R.

(ii) Let $e \in E$ and $S \in \mathfrak{S}$ such that the maximum for γ_1 is assumed there, i.e., $\gamma_1 = \frac{C_{eS}}{R_{eS}}$. Then for all $f \in E$ and $T \in \mathfrak{S}$ we have $\gamma_1 = \frac{C_{eS}}{R_{eS}} \geq \frac{C_{fT}}{R_{fT}}$, and so by the definition of γ_2 it follows $\frac{1}{\gamma_1} = \frac{R_{eS}}{C_{eS}} \leq \frac{R_{fT}}{C_{fT}} \leq \gamma_2$. This yields $1 \leq \gamma_1\gamma_2$.

(iii) Immediately clear. □

Now we turn to important global measures that depend on one or two action distributions. We already know two measures on strategy level, namely the *social cost* and the *mixed social cost*:

$$\mathsf{SC}(a) := \sum_{S\in\mathfrak{S}} L_S(a)a_S \qquad \text{for } a \in \mathcal{A}$$

$$\mathsf{SC}^a(b) := \sum_{S\in\mathfrak{S}} L_S(a)b_S \qquad \text{for } a, b \in \mathcal{A}$$

With NCGs, we often made use of that the social cost can also be expressed on element level, cf. (1.2). This is no longer true for NCRCGs in general. Therefore, we need separate measures on *element level*. These are the following, called *total element cost* and *mixed total element cost*, respectively:

$$\mathsf{EC}(a) \coloneqq \sum_{e \in E} \ell_e(g_e(a)) g_e(a) \qquad \text{for } a \in \mathcal{A}$$
$$\mathsf{EC}^a(b) \coloneqq \sum_{e \in E} \ell_e(g_e(a)) g_e(b) \qquad \text{for } a, b \in \mathcal{A}$$

If $C = R$, i.e., if we have an NCG, then $\mathsf{EC}^a(b) = \mathsf{SC}^a(b)$, and so $\mathsf{EC}(a) = \mathsf{SC}(a)$, but not in the general case.

2.6 Modeling Multicast Routing

We have seen in Sec. 2.3 that we can model two forms of selfish multicast routing easily as NCGs. We already sketched how the remaining two forms, namely duplication flows combined with path-based latency and conservation flows combined with edge-based latency, could be modeled as NCRCGs. We repeat this in detail here and also revisit the example from Sec. 2.2 as an illustration. Proper settings for matrices C and R in general are given in Tab. 2.1 on the next page. As in the proof of Thm. 2.1, we abuse notation by using links and strategies (in the sense of sets of paths) as matrix indices.

As an example, consider the multicast instance from Fig. 2.1 on p. 41. We use duplication flows combined with path-based latency. Matrices C and R are as follows. Recall that strategies correspond to columns, and elements correspond to rows. We have

$$C = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad R = \begin{pmatrix} 2 & 0 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix} .$$

Table 2.1. How to set consumption and relevance numbers in four variants of multicast. We have to set a consumption and a relevance number for each pair of an edge $e \in E$ and a strategy S.

	edge-based latency	path-based latency
conservation flow	$C_{eS} := \lvert S(e)\rvert$ $R_{eS} := \begin{cases} 1 & \text{if } e \in E(S) \\ 0 & \text{otherwise} \end{cases}$	$C_{eS} := \lvert S(e)\rvert$ $R_{eS} := \lvert S(e)\rvert$
duplication flow	$C_{eS} := \begin{cases} 1 & \text{if } e \in E(S) \\ 0 & \text{otherwise} \end{cases}$ $R_{eS} := \begin{cases} 1 & \text{if } e \in E(S) \\ 0 & \text{otherwise} \end{cases}$	$C_{eS} := \begin{cases} 1 & \text{if } e \in E(S) \\ 0 & \text{otherwise} \end{cases}$ $R_{eS} := \lvert S(e)\rvert$

It follows that $\gamma_1 = 1$ and $\gamma_2 = 2$. The price of anarchy was found to be at least $\Omega(2^p) = \Omega(\gamma_2^p)$ for $\theta = O(2^{1-p})$. For the example in [9] with k terminals, for a number $k \in \mathbb{N}$, we have $\gamma_2 = k$ and a lower bound of $\Omega(k^p) = \Omega(\gamma_2^p)$.

Parameters γ_1 and γ_2 are network-related. If combining conservation flows with edge-based latency, γ_1 gives the maximum number of terminals that are served by one edge, the maximum taken over all edges and strategies. For duplication flows with path-based strategy, γ_2 takes this role. So, in some sense, these parameters give the maximum 'bottleneck' role that an edge can have. The complementary parameters, γ_2 and γ_1, respectively, are 1 since we assume the terminals to be all distinct, which implies that there is an edge that only serves a single terminal. Without this assumption, these parameters may be lesser than 1, the extreme case being that all terminals are on the same vertex, resulting in $\gamma = \gamma_1\gamma_2 = 1$, which is not surprising, since such a configuration is essentially unicast.

2.7 Reducing γ by Scaling

Let $\varepsilon \in \mathbb{R}_{>0}$ and $e \in E$. If we scale that row of R corresponding to e with ε and then replace ℓ_e by $\tilde{\ell}_e(x) := \frac{1}{\varepsilon}\ell_e(x)$, essentially nothing changes for the game. Functions $\vec{L}$ and SC are unchanged, and hence so is the set of equilibria, optima, and their social costs. Also the β parameter does not change; if $\tilde{\ell}_e$ extends the class of used element latency functions from $\mathcal{L}$ to $\widetilde{\mathcal{L}}$, we have $\beta(\mathcal{L}) = \beta(\widetilde{\mathcal{L}})$ by Prop. 1.24. (The function EC changes, however.) Many of the upper bounds to be established later will be non-decreasing in the term $\gamma = \gamma_1\gamma_2$ and hence it is desirable to reduce that quantity, while preserving other important properties of the game. We explain in the following how to do so by scaling.

We call a vector $\vec{\varepsilon} = (\varepsilon_e)_{e\in E} \in \mathbb{R}^m_{>0}$ a *relevance scaling* (or here just *scaling*) and define $R^{\vec{\varepsilon}}$ to be the matrix R where row e was multiplied by ε_e for each $e \in E$. We would like to find an optimal scaling, i.e., one for that γ, taken with respect to $R^{\vec{\varepsilon}}$, is minimal over all possible scalings. It would be no extension to also consider series of scalings, since the sequential application of two scalings $\vec{\varepsilon}$ and $\vec{\delta}$ can as well be expressed by the one scaling $(\varepsilon_e\delta_e)_{e\in E}$. We call an instance *scalable* if it has a scaling that strictly reduces γ. So, if an instance $\tilde{\Gamma}$ is *not* scalable, then there is no way to strictly reduce its γ by means of a scaling. If a non-scalable $\tilde{\Gamma}$ is the result of scaling an instance Γ with $\vec{\varepsilon}$, then that scaling is optimal for Γ. For, if $\vec{\delta}$ was a better scaling for Γ, then the scaling $(\varepsilon_e^{-1}\delta_e)_{e\in E}$ would reduce γ of the non-scalable $\tilde{\Gamma}$, a contradiction.

We introduce notation for scaled 'γ'-values:

$$\gamma_1^{\vec{\varepsilon}}(e) := \max_{S\in\mathfrak{S}} \frac{C_{eS}}{R^{\vec{\varepsilon}}_{eS}} = \max_{S\in\mathfrak{S}} \frac{C_{eS}}{\varepsilon_e R_{eS}} \qquad \text{for } e \in E$$

$$\gamma_2^{\vec{\varepsilon}}(e) := \max_{S\in\mathfrak{S}} \frac{R^{\vec{\varepsilon}}_{eS}}{C_{eS}} = \max_{S\in\mathfrak{S}} \frac{\varepsilon_e R_{eS}}{C_{eS}} \qquad \text{for } e \in E$$

and

$$\gamma_1^{\vec{\varepsilon}} := \max_{e\in E} \gamma_1^{\vec{\varepsilon}}(e)\ , \qquad \gamma_2^{\vec{\varepsilon}} := \max_{e\in E} \gamma_2^{\vec{\varepsilon}}(e)\ , \qquad \gamma^{\vec{\varepsilon}} := \gamma_1^{\vec{\varepsilon}}\gamma_2^{\vec{\varepsilon}}\ .$$

Algorithm 1: Construction of an optimal relevance scaling.

$\vec{\varepsilon} \leftarrow (1, \ldots, 1)$;
while $\nexists e \in E$ *such that* $\gamma_1^{\vec{\varepsilon}} = \gamma_1^{\vec{\varepsilon}}(e)$ *and* $\gamma_2^{\vec{\varepsilon}} = \gamma_2^{\vec{\varepsilon}}(e)$ **do**
  $e^* \leftarrow \arg\max_{e \in E} \gamma_2^{\vec{\varepsilon}}(e)$;
  $\varepsilon_{e^*} \leftarrow \varepsilon_{e^*} \cdot \dfrac{\gamma_1^{\vec{\varepsilon}}(e^*)}{\gamma_1^{\vec{\varepsilon}}}$;
return $\vec{\varepsilon}$;

Then, clearly, we have a simple relation to the known quantities $\gamma_1 = \gamma_1^{(1,\ldots,1)}$ and $\gamma_2 = \gamma_2^{(1,\ldots,1)}$. Denote $\gamma_1(e) := \gamma_1^{(1,\ldots,1)}(e)$ and $\gamma_2(e) := \gamma_2^{(1,\ldots,1)}(e)$ for each $e \in E$. (Strictly, we should write "$(1,\ldots,1)^\top$" in the exponent, but we refrain from this for the sake of a more readable notation.)

2.7 Proposition. *If the maximum for γ_1 and the maximum for γ_2 are both attained for the same element, then the instance is not scalable.*

Proof. Let $e \in E$ such that $\gamma_i = \gamma_i(e)$, $i = 1, 2$. Let $\vec{\varepsilon} \in \mathbb{R}^m_{>0}$ be a relevance scaling. We have $\gamma_1^{\vec{\varepsilon}}(e) = \frac{1}{\varepsilon_e}\gamma_1(e)$ and $\gamma_2^{\vec{\varepsilon}}(e) = \varepsilon_e \gamma_2(e)$. Since for $\gamma_1^{\vec{\varepsilon}}$ and $\gamma_2^{\vec{\varepsilon}}$, the maximum is taken over all elements, they cannot be smaller than $\gamma_1^{\vec{\varepsilon}}(e)$ and $\gamma_2^{\vec{\varepsilon}}(e)$, respectively. It follows $\gamma^{\vec{\varepsilon}} = \gamma_1^{\vec{\varepsilon}}\gamma_2^{\vec{\varepsilon}} \geq \frac{1}{\varepsilon_e}\gamma_1(e)\varepsilon_e\gamma_2(e) = \gamma_1(e)\gamma_2(e) = \gamma$. □

2.8 Proposition. *Algorithm 1 constructs an optimal relevance scaling in at most m iterations.*

Proof. By Prop. 2.7, the stopping criterion ensures that upon termination, the given relevance scaling is optimal. Hence it is left to show that the procedure stops after at most m iterations.

Consider an arbitrary iteration. Let T^* be such that the maximum for $\gamma_1^{\vec{\varepsilon}}(e^*)$ is attained in T^*, i.e., $\gamma_1^{\vec{\varepsilon}}(e^*) = \frac{C_{e^*T^*}}{R^{\vec{\varepsilon}}_{e^*T^*}}$. The scaling $\vec{\varepsilon}$ is replaced by a new one, say $\vec{\delta}$, such that by the choice of T^* we have

$$\gamma_1^{\vec{\delta}}(e^*) = \max_{T \in \mathfrak{S}} \frac{C_{e^*T}}{R^{\vec{\delta}}_{e^*T}} = \max_{T \in \mathfrak{S}} \frac{C_{e^*T}}{R^{\vec{\varepsilon}}_{e^*T} \frac{C_{e^*T^*}}{R^{\vec{\varepsilon}}_{e^*T^*}} \frac{1}{\gamma_1^{\vec{\varepsilon}}}} = \frac{R^{\vec{\varepsilon}}_{e^*T^*}}{C_{e^*T^*}} \gamma_1^{\vec{\varepsilon}} \max_{T \in \mathfrak{S}} \frac{C_{e^*T}}{R^{\vec{\varepsilon}}_{e^*T}} = \gamma_1^{\vec{\varepsilon}} \ .$$

In other words, the new scaling is so that $\gamma_1^{\vec{\delta}}(e^*)$ just not exceeds $\gamma_1^{\vec{\varepsilon}}$, and this makes $\gamma_1^{\vec{\delta}}(e^*) = \gamma_1^{\vec{\varepsilon}}$. As a result, note that the 'γ_1'-value does not change during the whole procedure. So the row is in fact left with $\gamma_1^{\vec{\delta}}(e^*) = \gamma_1$.

No row will be treated twice, for if a row e^* was chosen a second time, we would on the one hand have $\gamma_1^{\vec{\varepsilon}} = \gamma_1 = \gamma_1^{\vec{\varepsilon}}(e^*)$ as a result of a previous iteration. On the other hand, by the selection criterion for e^*, also $\gamma_2^{\vec{\varepsilon}} = \gamma_2^{\vec{\varepsilon}}(e^*)$. It follows that the procedure would have stopped before that. Hence, there are at most m iterations. □

We make some remarks to illustrate further how the algorithm works. In each iteration, the row responsible for $\gamma_2^{\vec{\varepsilon}}$ is scaled down as much as possible without spoiling γ_1. The row is indeed scaled *down*, since the additional scaling factor $\frac{\gamma_1^{\vec{\varepsilon}}(e^*)}{\gamma_1^{\vec{\varepsilon}}}$ is strictly smaller than 1 by the properties of $\gamma_1^{\vec{\varepsilon}}$: by definition $\gamma_1^{\vec{\varepsilon}}(e^*) \leq \gamma_1^{\vec{\varepsilon}}$, and since γ_1^{ε} is not attained in row e^*, this inequality holds strict. Since no row is treated twice, we could as well have written "$\varepsilon_{e^*} \leftarrow \frac{\gamma^{\vec{\varepsilon}}(e^*)}{\gamma_1^{\vec{\varepsilon}}}$" in the algorithm. Now, there are two possible outcomes of scaling row e^*. Either is $\gamma_2^{\vec{\delta}}$ still attained for e^* and the procedure terminates after this iteration. Or the row in which $\gamma_2^{\vec{\delta}}$ is attained has changed to some $f \neq e^*$. Since only row e^* is modified when switching from $\vec{\varepsilon}$ to $\vec{\delta}$, we have $\gamma_2^{\vec{\delta}}(f) \leq \gamma_2^{\vec{\varepsilon}}$.

Scaling rows of the consumption matrix C is also possible. If a row e of C is scaled by $\delta \in \mathbb{R}_{>0}$, then we have to replace ℓ_e by $x \mapsto \ell_e(\frac{1}{\delta}x)$. However, for the sake of reducing $\gamma = \gamma_1\gamma_2$ it does not matter whether we scale rows of R, or of C, or of both. For, the combination of a relevance scaling $\vec{\varepsilon}$ and consumption scaling $\vec{\delta}$ can be transformed into an equivalent relevance scaling: set $\lambda_e := \varepsilon_e/\delta_e$ for each $e \in E$. Then scaling R with $\vec{\lambda}$ has the same effect on γ as the combination of $\vec{\varepsilon}$ used on R and $\vec{\delta}$ used on C.

2.8 General Lower Bound on the Price of Anarchy

Using the construction from [9], we can have multicast instances with a price of anarchy at least $\Omega(\gamma_2^p)$ for any desired $\gamma_2 \in \mathbb{N}_{\geq 1}$, using

element latency functions from $\mathrm{Poly}^+(p)$. We can extend that using the flexibility of general NCRCGs and also give a special bound for small γ.

2.9 Theorem.

(i) Let $p \in \mathbb{N}_{\geq 1}$ and $c, r \in \mathbb{R}_{\geq 1}$. There exist non-scalable NCRCGs with element latency functions only from $\mathrm{Poly}^+(p)$ with $\gamma_1 = c$ and $\gamma_2 = r$ such that the price of anarchy is at least $\gamma^p = (\gamma_1\gamma_2)^p$.

(ii) Let $p \in \mathbb{N}_{\geq 1}$ and $c, r \in \mathbb{R}_{\geq 1}$ such that $cr < (p+1)^{\frac{1}{p}}$. There exist non-scalable NCRCGs with element latency functions only from $\mathrm{Poly}^+(p)$ with $\gamma_1 = c$ and $\gamma_2 = r$ such that the price of anarchy is at least

$$\frac{1}{1-\gamma\beta}\ , \quad \text{where } \beta = \beta(\mathrm{Poly}^+(p)) = p\,(p+1)^{-1-\frac{1}{p}}.$$

Both lower bounds from Thm. 2.9 have an upper-bound counterpart, which is no more than a factor of γ away and will be given in Thm. 2.12.

Proof of Thm. 2.9. (i) Different kinds of instances can be used to show this worst-case lower bound. We give one that, provided $c, r \in \mathbb{N}$, only uses integral numbers in C and R, and only integral coefficients in the polynomials. Let $N := 1$, $d := 1$, $m := r$, and $n := m + 1$ and define:

$$\begin{aligned} C &:= \begin{pmatrix} 1 & & & c \\ & \ddots & & \vdots \\ & & 1 & c \end{pmatrix} \in \mathbb{R}^{m\times(m+1)} \\ R &:= \begin{pmatrix} r & & & 1 \\ & \ddots & & \vdots \\ & & r & 1 \end{pmatrix} \in \mathbb{R}^{m\times(m+1)} \end{aligned} \tag{2.3}$$

Empty entries are 0. Then $\gamma_1 = c$ and $\gamma_2 = r$, and the instance is not scalable. Let each element latency function be $x \mapsto x^p$ and $a := (0, \ldots, 0, 1)$ and $a^* := (\frac{1}{m}, \ldots, \frac{1}{m}, 0)$. Then a is a Nash equilibrium, since

$$\forall S \in [m] \quad L_S(a) = rc^p \quad \text{and}$$

$$L_{m+1}(a) = \sum_{e=1}^{m} c^p = mc^p = rc^p \ .$$

We have its social cost

$$\mathsf{SC}(a) = rc^p d = rc^p \ . \tag{2.4}$$

For a^* we have the following (showing as a side-product that it happens to be a Nash equilibrium as well):

$$\forall S \in [m] \quad L_S(a^*) = r\left(\frac{1}{m}\right)^p = \left(\frac{1}{r}\right)^{p-1} \quad \text{and}$$
$$L_{m+1}(a^*) = \sum_{e=1}^{m}\left(\frac{1}{m}\right)^p = m\left(\frac{1}{m}\right)^p = \left(\frac{1}{r}\right)^{p-1} \ .$$

We have its social cost

$$\mathsf{SC}(a^*) = \left(\frac{1}{r}\right)^{p-1} d = \left(\frac{1}{r}\right)^{p-1} \ .$$

It follows that $\mathsf{SC}(a)/\mathsf{SC}(a^*) = rc^p\, r^{p-1} = (cr)^p = (\gamma_1\gamma_2)^p = \gamma^p$. This proves (i).

(ii) We extend the example from (i) by a strategy with index $r+2$ with constant latency $L_{r+2}(\cdot) = rc^p$. This can be implemented by an additional element and the following matrices:

$$C \coloneqq \begin{pmatrix} 1 & & & c \\ & \ddots & & \vdots \\ & & 1 & c \\ & & & 1 \end{pmatrix} \in \mathbb{R}^{(m+1)\times(m+2)}$$

$$R \coloneqq \begin{pmatrix} r & & & 1 \\ & \ddots & & \vdots \\ & & r & 1 \\ & & & 1 \end{pmatrix} \in \mathbb{R}^{(m+1)\times(m+2)}$$

This instance is not scalable. The additional element (corresponding to the last row) is given constant element latency function $x \mapsto rc^p$.

By essentially the same argument as before, $a := (0, \dots, 0, 1, 0)$ is a Nash equilibrium with social cost $\mathsf{SC}(a) = rc^p$, just note that the newly introduced strategy does not offer better latency than the other ones under a. Define furthermore

$$a^* := \left(\frac{c}{(p+1)^{1/p}}, \dots, \frac{c}{(p+1)^{1/p}}, 0, 1 - r\frac{c}{(p+1)^{1/p}} \right) \in \mathbb{R}^{r+2}_{\geq 0} \ .$$

Then by assumption on cr, the vector a^* is an action distribution. So we follow the idea of the first lower bound, but additionally take advantage of the fact that we can move some of the demand to the new strategy with constant latency.[2] The social cost of a^* is

$$\begin{aligned}
\mathsf{SC}(a^*) &= \sum_{S=1}^{r} L_S(a^*) a^*_S + L_{r+2}(a^*) a^*_{r+2} \\
&= \sum_{S=1}^{r} r \frac{c^p}{p+1} \frac{c}{(p+1)^{1/p}} + rc^p \left(1 - r\frac{c}{(p+1)^{1/p}}\right) \\
&= rc^p \left(r \frac{1}{p+1} \frac{c}{(p+1)^{1/p}} + \left(1 - r\frac{c}{(p+1)^{1/p}}\right)\right) \\
&= rc^p \left(1 - rc\left((p+1)^{-\frac{1}{p}} - (p+1)^{-1}(p+1)^{-\frac{1}{p}}\right)\right) \\
&= rc^p \left(1 - rc\left((p+1)(p+1)^{-1-\frac{1}{p}} - (p+1)^{-1-\frac{1}{p}}\right)\right) \\
&= rc^p \left(1 - rc\, p\, (p+1)^{-1-\frac{1}{p}}\right) \ .
\end{aligned}$$

Recall that Nash equilibrium a has $\mathsf{SC}(a) = rc^p$. Hence

$$\frac{\mathsf{SC}(a)}{\mathsf{SC}(a^*)} = \frac{1}{1 - rc\, p\, (p+1)^{-1-\frac{1}{p}}} = \frac{1}{1 - \gamma\, p\, (p+1)^{-1-\frac{1}{p}}} = \frac{1}{1-\gamma\beta} \ . \quad \square$$

2.9 Upper Bound on the Price of Anarchy

Recall the characterization of Nash equilibria given in Thm. 1.6. We can apply that to prove upper bounds on the price of anarchy, inspired

[2] The exact setting was obtained by introducing a parameter x, then defining $a^*(x) := (x, \dots, x, 0, 1 - rx)$, and finally choosing an x with $\mathsf{SC}(a^*(x))$ minimal under the constraint that $a^*(x)$ is an action distribution.

by the basic idea from the proof of Thm. 1.18. We first relate (mixed) total element cost to (mixed) social cost.

2.10 Proposition. *Let $a, b \in \mathcal{A}$. Then we have*

$$\frac{1}{\gamma_2}\mathsf{SC}^a(b) \le \mathsf{EC}^a(b) \le \gamma_1 \mathsf{SC}^a(b) \ .$$

Proof. We start with the first inequality. We use the convention $\frac{0}{0} := 0$.

$$\begin{aligned}\mathsf{SC}^a(b) &= \sum_{S\in\mathfrak{S}} L_S(a) b_S = \sum_{S\in\mathfrak{S}}\sum_{e\in E} R_{eS}\ell_e(g_e(a))b_S \\ &= \sum_{S\in\mathfrak{S}}\sum_{e\in E}\frac{R_{eS}}{C_{eS}} C_{eS}\ell_e(g_e(a))b_S \le \gamma_2 \sum_{S\in\mathfrak{S}}\sum_{e\in E} C_{eS}\ell_e(g_e(a))b_S \\ &= \gamma_2 \sum_{e\in E}\ell_e(g_e(a)) \underbrace{\sum_{S\in\mathfrak{S}} C_{eS} b_S}_{=g_e(b)} = \gamma_2 \mathsf{EC}^a(b) \ .\end{aligned}$$

The second inequality follows likewise:

$$\begin{aligned}\mathsf{EC}^a(b) &= \sum_{e\in E}\ell_e(g_e(a))g_e(b) = \sum_{e\in E}\ell_e(g_e(a)) \sum_{S\in\mathfrak{S}} C_{eS} b_S \\ &= \sum_{S\in\mathfrak{S}}\sum_{e\in E}\frac{C_{eS}}{R_{eS}} R_{eS}\ell_e(g_e(a))b_S \\ &\le \gamma_1 \sum_{S\in\mathfrak{S}} b_S \underbrace{\sum_{e\in E} R_{eS}\ell_e(g_e(a))}_{=L_S(a)} = \gamma_1 \mathsf{SC}^a(b) \ .\end{aligned}$$

□

For the rest of this section, we will treat NCRCGs with super-homogeneous element latency functions. Recall the definition of super-homogeneous element latency functions and their properties from Sec. 1.7.2. An element latency function ℓ is s-super-homogeneous if $\ell(\varepsilon x) \ge s(\varepsilon)\ell(x)$ for all $\varepsilon \in (0,1]$ and $x \in \mathbb{R}_{\ge 0}$. This is equivalent to being $\bar{s}$-sub-homogeneous, i.e., $\ell(tx) \le \bar{s}(t)\ell(x)$ for all $t \in [1,\infty)$ and $x \in \mathbb{R}_{\ge 0}$ with $\bar{s}(t) = s(t^{-1})^{-1}$. A class $\mathcal{L}$ of element latency functions is called s-super-homogeneous if each $\ell \in \mathcal{L}$ is s-super-homogeneous. For instance, class $\mathrm{Poly}^+(p)$ is s-super-homogeneous for each p with $s(\varepsilon) = \varepsilon^p$, so it is also $\bar{s}$-sub-homogeneous with $\bar{s}(t) = t^p$.

2.11 Lemma. *Let $\mathcal{L}$ be an s-super-homogeneous class of element latency functions. Let an NCRCG be given with element latency functions drawn from $\mathcal{L}$. Let $a, b \in \mathbb{R}^n_{\geq 0}$ and $t \geq 1$. Then*

$$\mathsf{EC}^a(b) \leq \frac{1}{t}\beta(\mathcal{L})\mathsf{EC}(a) + \bar{s}(t)\mathsf{EC}(b) \ .$$

Proof.

$$\begin{aligned}
&t\,\mathsf{EC}^a(b) = \sum_{e\in E} \ell_e(g_e(a))\, t g_e(b) \\
&= \sum_{e\in E} \big(\ell_e(g_e(a)) - \ell_e(t g_e(b)))t g_e(b) + t\ell_e(t g_e(b)) g_e(b)\big) \\
&\leq \sum_{e\in E} \big(\beta(\mathcal{L})\ell_e(g_e(a)) g_e(a) + t\ell_e(t g_e(b)) g_e(b)\big) \quad \text{(1.11) on p. 25} \\
&\leq \sum_{e\in E} \big(\beta(\mathcal{L})\ell_e(g_e(a)) g_e(a) + t\bar{s}(t)\ell_e(g_e(b)) g_e(b)\big) \quad \bar{s}\text{-sub-homogeneity} \\
&= \beta(\mathcal{L})\mathsf{EC}(a) + t\bar{s}(t)\mathsf{EC}(b) \ .
\end{aligned}$$

Dividing by t yields the claimed inequality. □

The following is our main result in this section.

2.12 Theorem.

(i) The price of anarchy in an NCRCG with element latency functions drawn from an s-super-homogeneous class $\mathcal{L}$ for which $\beta := \beta(\mathcal{L}) < 1$ holds is no more than

$$\begin{cases} \frac{1}{1-\beta}\gamma\bar{s}(\gamma) & \text{for all } \gamma \\ \min\left\{\frac{1}{1-\gamma\beta}\gamma,\ \frac{1}{1-\beta}\gamma\bar{s}(\gamma)\right\} & \text{for } \gamma < 1/\beta \end{cases} .$$

(ii) The price of anarchy in an NCRCG with element latency functions drawn from $\mathrm{Poly}^+(p)$ is no more than

$$\begin{cases} \gamma^{p+1} & \text{if } \gamma \geq (1+p)^{\frac{1}{p}} \\ \frac{1}{1-\gamma\beta}\gamma & \text{if } \gamma \leq (1+p)^{\frac{1}{p}} \end{cases} ,$$

where $\beta = \beta(\mathrm{Poly}^+(p)) = p\,(p+1)^{-1-\frac{1}{p}}$.

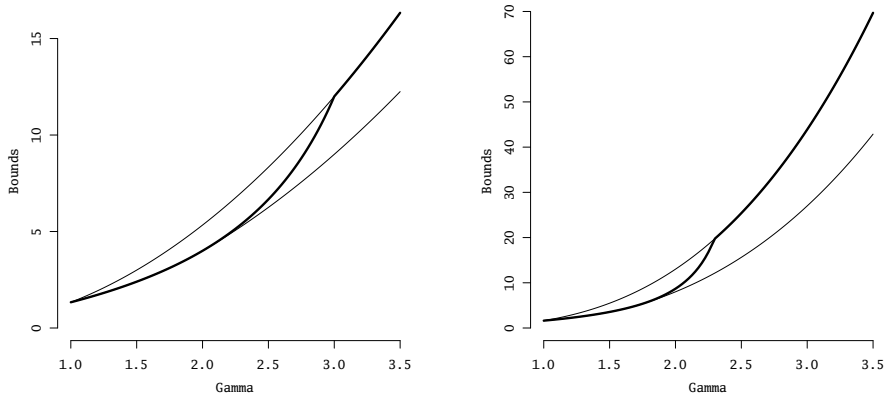


Figure 2.4. Comparison of three different bounds for polynomial latency functions, $p = 1$ on the left and $p = 2$ on the right, and γ in the range between 1 and 3.5. The universal bound, using $\bar{s}(\gamma) = \gamma^p$, upper-bounds the other two. The bound specifically for polynomials, i.e., Thm. 2.12(ii), is the best one for all γ. The refined version of the universal bound, i.e., the second bound in Thm. 2.12(i), first follows the specific bound, then it is in between for some range of γ, and finally it follows the universal bound; it is plotted thicker than the other two.

Let us briefly discuss these bounds before starting the proof. The bound $\frac{1}{1-\beta}\gamma\bar{s}(\gamma)$ is universal; it holds for all γ and all classes $\mathcal{L}$ of element latency functions, provided only that $\beta = \beta(\mathcal{L}) < 1$. The bound $\frac{1}{1-\gamma\beta}\gamma$ is only proved for small γ and is sometimes better than the universal one, depending on the function $\bar{s}$. The second part of the theorem treats the case of polynomial element latency functions, i.e., from $\text{Poly}^+(p)$ for a fixed p. We distinguish between large and small γ. Large γ start at $(1+p)^{\frac{1}{p}}$, and it will become clear from the proof why we chose that threshold. The bound for large γ improves on the universal one by a factor of $\frac{1}{1-\beta}$. The bound for small γ is a corollary from the bounds in part (i) of the theorem. In fact we get a bound of $\min\left\{\frac{1}{1-\gamma\beta}\gamma,\ \frac{1}{1-\beta}\gamma^{p+1}\right\}$. However, real calculus shows that the first expression is always upper-bounded by the second one and hence the first is the better bound. Fig. 2.4 gives an illustration.

Proof of Thm. 2.12. Let a be a Nash equilibrium and b be any action distribution. For all $t \geq 1$ we have

$$\begin{aligned}
\mathsf{SC}(a) &\leq \mathsf{SC}^a(b) \leq \gamma_2 \mathsf{EC}^a(b) && \text{by Thm. 1.6 and Prop. 2.10} \\
&\leq \gamma_2 \left(\frac{1}{t} \beta \mathsf{EC}(a) + \bar{s}(t) \mathsf{EC}(b) \right) && \text{by Lem. 2.11} \\
&\leq \gamma_2 \gamma_1 \left(\frac{1}{t} \beta \mathsf{SC}(a) + \bar{s}(t) \mathsf{SC}(b) \right) && \text{by Prop. 2.10} \\
&= \gamma \left(\frac{1}{t} \beta \mathsf{SC}(a) + \bar{s}(t) \mathsf{SC}(b) \right) \ .
\end{aligned}$$

Key observation: if $t \geq 1$ is chosen such that $1 - \gamma \frac{1}{t} \beta > 0$, the above yields

$$\frac{\mathsf{SC}(a)}{\mathsf{SC}(b)} \leq \frac{1}{1 - \gamma \frac{1}{t} \beta} \gamma \bar{s}(t).$$

All bounds follow from the key observation by an appropriate choice of t:

- For the first bound of part (i), i.e., the universal bound, choose $t \coloneqq \gamma$ and note that $t \geq 1$ holds then, as well as $1 - \gamma \frac{1}{t} \beta = 1 - \beta > 0$ since $\beta < 1$.
- To prove the second bound of part (i), let $\gamma < 1/\beta$, hence $\gamma\beta < 1$. We only have to show the $\frac{1}{1-\gamma\beta}\gamma$ bound, since the other one is the universal bound. Choose $t \coloneqq 1$. Then $1 - \gamma \frac{1}{t} \beta = 1 - \gamma\beta > 0$, and the bound follows from the key observation, since $\bar{s}(1) = 1$.
- To prove the first bound from part (ii), choose $t \coloneqq \gamma \, (p+1)^{-\frac{1}{p}}$, which is ≥ 1 by assumption on γ, and also implies

$$1 - \gamma \frac{1}{t} \beta = 1 - (p+1)^{\frac{1}{p}} p \, (p+1)^{-1-\frac{1}{p}} = 1 - \frac{p}{p+1} > 0 \ .$$

The key observation yields

$$\frac{\mathsf{SC}(a)}{\mathsf{SC}(b)} \leq \frac{\gamma \, t^p}{1 - \frac{p}{p+1}} = \frac{\gamma \gamma^p (p+1)^{-1}}{1 - \frac{p}{p+1}} = \gamma^{p+1} \ .$$

– For the second bound from part (ii) we just have to note that

$$\gamma\beta \leq (p+1)^{\frac{1}{p}} p\,(p+1)^{-1-\frac{1}{p}} = \frac{p}{p+1} < 1 \ ,$$

hence $\gamma < 1/\beta$, and invoke the second bound from part (i). □

2.13 Remark. Bounds in Thm. 2.12(ii) are tight up to a factor of γ by Thm. 2.9.

When establishing the upper bound from Thm. 2.12 for a concrete instance with non-decreasing $\bar{s}$ (as it is the case for polynomials), we can first scale the instance in order to minimize γ, as described in Sec. 2.7. If scaling extends the class of element latency functions occurring from $\mathcal{L}$ to some $\widetilde{\mathcal{L}}$, we have $\beta(\mathcal{L}) = \beta(\widetilde{\mathcal{L}})$ by Prop. 1.24. Since all bounds are non-increasing in γ and, for part (ii), we have $\frac{1}{1-\gamma\beta}\gamma = \gamma^{p+1}$ if $\gamma = (1+p)^{\frac{1}{p}}$, the bounds can only improve by the scaling.

The bounds for the scaled instance also hold for the original one. So, in particular, it is not necessary to 'really' scale the instance – if, e.g., further algorithmic tasks shall be carried out with the instance, we can well work with the original one. This is how we will proceed in our experimental studies in Ch. 3.

2.10 Bicriteria Bound

The next theorem says that a Nash equilibrium is not worse in terms of social cost than an optimum for $(1+\beta(\mathcal{L}))\,\gamma$ times the demand. It is hence a natural extension to the known analogous result for NCGs, which uses a scaling of the demand by $1+\beta(\mathcal{L})$.

2.14 Theorem. *Let Γ be an NCRCG with element latency functions drawn from the s-super-homogeneous class $\mathcal{L}$. Let a be a Nash equilibrium for Γ and b any action distribution for that NCRCG resulting from Γ by a multiplication of each demand d_i by $(1+\beta(\mathcal{L}))\,\gamma$. Then $\mathsf{SC}(a) \leq \mathsf{SC}(b)$.*

Proof. Let $t, u \geq 1$ to be specified more precisely later. Let b be an action distribution for the demands scaled up by u. Then $\frac{1}{u}b$ is an

action distribution for Γ and hence

$$\begin{aligned}
\mathsf{SC}(a) &\le \mathsf{SC}^a(\frac{1}{u}b) \le \gamma_2 \mathsf{EC}^a(\frac{1}{u}b) && \text{by Thm. 1.6 and Prop. 2.10}\\
&= \gamma_2 \frac{1}{u} \mathsf{EC}^a(b)\\
&\le \gamma_2 \frac{1}{u}(\frac{1}{t}\beta(\mathcal{L})\mathsf{EC}(a) + \bar{s}(t)\mathsf{EC}(b)) && \text{by Lem. 2.11}\\
&\le \gamma \frac{1}{u}(\frac{1}{t}\beta(\mathcal{L})\mathsf{SC}(a) + \bar{s}(t)\mathsf{SC}(b)) \ . && \text{by Prop. 2.10}
\end{aligned}$$

It follows that

$$\mathsf{SC}(a) \underbrace{\left(\frac{u}{\gamma} - \frac{\beta(\mathcal{L})}{t}\right)\frac{1}{\bar{s}(t)}}_{\phi(u,t) :=} \le \mathsf{SC}(b) \ .$$

To receive "$\mathsf{SC}(a) \le \mathsf{SC}(b)$", we need that $\phi(u,t) \ge 1$. Since $\phi(\cdot,t)$ is strictly increasing and we wish to have u as small as possible, we aim for $\phi(u,t) = 1$, i.e.,

$$u = \left(\bar{s}(t) + \frac{\beta(\mathcal{L})}{t}\right)\gamma \ . \tag{2.5}$$

Choosing $t := 1$, we are forced to choose $u := (1+\beta(\mathcal{L}))\,\gamma$ and get (2.5); recall that $\bar{s}(1) = 1$. Note that for the case $\bar{s}(t) = t^p$, this choice of t is optimal, since for such $\bar{s}$ the right-hand side in (2.5) increases with t. □

2.15 Theorem. *Consider element latency functions from* $\mathrm{Poly}^+(p)$. *The scaling factor* $(1+\beta(\mathcal{L}))\,\gamma$ *in the previous theorem is the best possible up to a factor of* $(1+\beta(\mathcal{L}))\,\gamma^{\frac{1}{p+1}}$.

Proof. Recall the first example from the proof of Thm. 2.9, displayed in (2.3) on p. 55. We have parameters $c, r, p \in \mathbb{N}_{\ge 1}$, $m = r$, $n = m+1$, element latency functions $x \mapsto x^p$, and as shown in that proof, we can construct a Nash equilibrium a with social cost $\mathsf{SC}(a) = rc^p$, see (2.4). On the other hand, if we scale up the demand of 1 by factor of $u \ge 1$, then there is an action distribution $b := (\frac{u}{m}, \ldots, \frac{u}{m}, 0) \in \mathbb{R}^{m+1}_{\ge 0}$. Let us

compute $\mathsf{SC}(b)$. We have

$$\forall S \in [m] \quad L_S(b) = r\left(\frac{u}{m}\right)^p = r\left(\frac{u}{r}\right)^p \quad \text{and}$$
$$L_{m+1}(b) = \sum_{e=1}^{m}\left(\frac{u}{m}\right)^p = m\left(\frac{u}{m}\right)^p = r\left(\frac{u}{r}\right)^p \ .$$

It follows that $\mathsf{SC}(b) = r\left(\frac{u}{r}\right)^p u = r^{1-p}u^{p+1}$. This is strictly increasing in u. If we set

$$u \coloneqq \frac{(1+\beta(\mathcal{L}))\,\gamma}{(1+\beta(\mathcal{L}))\,\gamma^{\frac{1}{p+1}}} = \gamma^{1-\frac{1}{p+1}} = (cr)^{1-\frac{1}{p+1}} \ , \tag{2.6}$$

then

$$\mathsf{SC}(b) = r^{1-p}u^{p+1} = r^{1-p}(cr)^{(1-\frac{1}{p+1})\,(p+1)} = r^{1-p}(cr)^p = rc^p = \mathsf{SC}(a) \ .$$

Hence we have to scale the demands up by at least this much, i.e., by u as given in (2.6), in order to have $\mathsf{SC}(\tilde{b}) \geq \mathsf{SC}(a)$ for all action distributions $\tilde{b}$ that are admissible for the scaled demands (such as b). □

2.11 Computation

Non-linear programs play an important role for computation of optima and Nash equilibria. *Convex* programs can be solved in polynomial time up to an arbitrarily small error, given some mild additional properties. Some of the practical methods require certain boundedness conditions, smoothness and efficiently computable first- and second-order derivatives of the functions involved. See, e.g., [12, 17, 70] for comprehensive treatments of convex optimization. The case of a convex quadratic objective function and linear constraints is polynomially solvable with the ellipsoid algorithm [55] or, up to an arbitrarily small error, by interior-point algorithms [12, 17, 70]. We will encounter that special case later with affine element latency functions, since then SC is a quadratic function and can be convex in some cases.

Computation of Nash equilibria and optima is hindered by two changes that occur when we switch from NCGs to NCRCGs:

– SC is not necessarily a convex function anymore;

– there seems to be no way to use the KKT theorem to recognize Nash equilibria as optima of a convex program, as it can be done for NCGs [11, 18, 19, 37], see also Sec. 1.6.

We will address these concerns separately below. Before, we state a fact about EC. By Prop. 2.10, an algorithm that minimizes EC yields a γ-approximation for OPT. This is especially interesting if $x \mapsto \ell_e(x)x$ is a convex function for all $e \in E$, since then EC is a convex function, even a separable one if considered a function of the congestions. However, it is clear that such an approach can only be of limited use since the computation does not involve matrix R. For a more accurate treatment, we will show how to work on *strategy level* for optima and Nash equilibria in the rest of this section.

2.11.1 Convexity and Non-Convexity of SC and Computation of Optima

Let all element latency functions be twice continuously differentiable on $\mathbb{R}_{>0}$, which implies that SC is a twice continuously differentiable function on $\mathbb{R}^n_{>0}$ with its Hessian $\nabla^2 \mathsf{SC}(v)$ at $v \in \mathbb{R}^n_{>0}$ being

$$\left(\sum_{e \in E} \left\{ (R_{eS_1} C_{eS_2} + R_{eS_2} C_{eS_1}) \ell_e'(g_e(v)) + \sum_{S \in \mathfrak{S}} R_{eS} C_{eS_1} C_{eS_2} \ell_e''(g_e(v)) v_S \right\} \right)_{\substack{S_1 \in \mathfrak{S} \\ S_2 \in \mathfrak{S}}}.$$

If all element latency functions are affine, say, $\ell_e(x) = \theta_e x + \tau_e$ with $\theta_e, \tau_e \in \mathbb{R}_{\geq 0}$, then $\nabla^2 \mathsf{SC}(v)$ is independent of v, namely

$$\nabla^2 \mathsf{SC}(v) = \left(\sum_{e \in E} \theta_e (R_{eS_1} C_{eS_2} + R_{eS_2} C_{eS_1}) \right)_{\substack{S_1 \in \mathfrak{S} \\ S_2 \in \mathfrak{S}}}. \tag{2.7}$$

A real-valued, twice differentiable function of multiple real variables is convex on a convex open set if and only if its Hessian is positive semidefinite in every point of that set. We cannot apply that directly

to SC with respect to $\mathcal{A}$, since $\mathcal{A}$ is not open (it is convex, though). If $\nabla^2 \mathsf{SC}$ is positive semidefinite in every point of some open convex set $U \subseteq \mathbb{R}^n_{>0}$ such that $\mathcal{A} \subseteq \overline{U}$, then SC is convex on U and hence, by continuity, also on $\mathcal{A}$. The converse, however, does not hold, and so a test based on this can deliver false positives, i.e., categorize SC as non-convex when, in fact, it is convex on $\mathcal{A}$. As an example, consider $C := \begin{pmatrix} 1 & 1 \end{pmatrix}$ and $R := \begin{pmatrix} 3 & 1 \end{pmatrix}$, let the one element have latency function $x \mapsto x$, and a demand $d = 1$ be given. Then, SC is convex on $\mathcal{A} = \{(a_1, a_2);\ a_1 + a_2 = 1\}$. However $\nabla^2 \mathsf{SC}(\cdot) = \begin{pmatrix} 6 & 4 \\ 4 & 2 \end{pmatrix}$ is not positive semidefinite; it has negative eigenvalue $4 - \sqrt{20}$.

For an exact test, we have to use the *projected Hessian* instead. Details are explained in the following. Let strategies be ordered such that strategies from one player class are grouped together and these groups arranged in the order $i = 1, \ldots, N$. Set $n_i := |\mathfrak{S}_i|$ for each $i \in [N]$. So, in our ordering the first n_1 strategies are from player class 1, the next n_2 strategies are from player class 2, etc. Let $\mathfrak{S}'_i$ be the set $\mathfrak{S}_i$ without the last strategy, so $|\mathfrak{S}'_i| = n_i - 1$. We will use an ordering on $\mathfrak{S}' := \bigcup_{i \in [N]} \mathfrak{S}'_i$ similar to that on $\mathfrak{S}$, i.e., the first $n_1 - 1$ strategies in $\mathfrak{S}'$ are from player class 1, the next $n_2 - 1$ strategies are from player class 2, etc. Define the affine mapping T and its matrix $M_T \in \mathbb{R}^{n \times (n-N)}$ as shown in Fig. 2.5 on the facing page. The version for $N = 1$ is as follows, empty entries in the matrix are zero:

$$T : \mathbb{R}^{n-1} \longrightarrow \mathbb{R}^n, (v_1, \ldots, v_{n-1})^\top \mapsto \underbrace{\begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \\ -1 & -1 & \ldots & -1 \end{pmatrix}}_{M_T :=} v + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ d \end{pmatrix} .$$

For each $i \in [N]$ we have the following: $n_i - 1$ real numbers are mapped by T to n_i real numbers such that the sum of the latter is

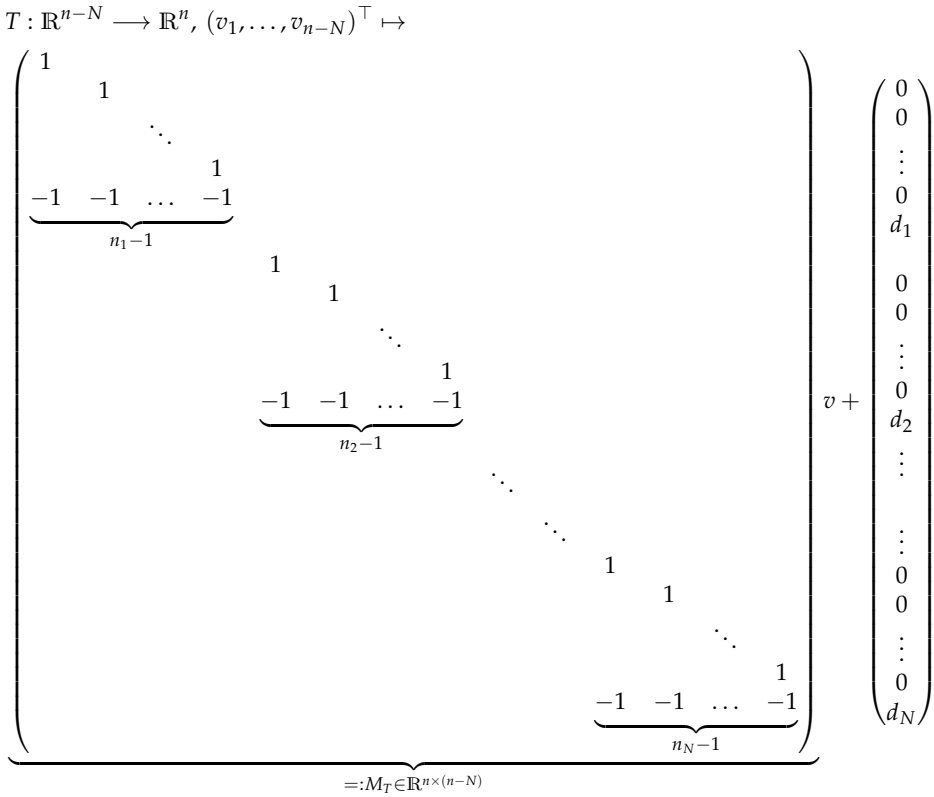


Figure 2.5. The affine mapping T. Empty entries in the matrix are zero.

always d_i. Define the set of *reduced action distributions*

$$\mathcal{A}' \coloneqq \{v \in \mathbb{R}^{n-N}_{>0};\ \sum_{S \in \mathfrak{S}'_i} v_S < d_i \quad \forall i \in [N]\}\ .$$

This is an open convex set in $\mathbb{R}^{n-N}$. The image of $\mathcal{A}'$ under T is contained in $\mathbb{R}^n_{>0}$ and it consists of all action distributions save those which have one or more zero entries. It follows that $\overline{T(\mathcal{A}')} = \mathcal{A}$. By continuity of SC, we hence know that SC is convex on $T(\mathcal{A}')$ if and only if it is convex on $\mathcal{A}$. Since T is an affine mapping, SC is convex on $T(\mathcal{A}')$ if and only if $\mathsf{SC} \circ T$ is convex on $\mathcal{A}'$. The latter is the case if and

only if $\nabla^2(\mathsf{SC} \circ T)(v)$ is positive semidefinite for all $v \in \mathcal{A}'$, since $\mathcal{A}'$ is open.

We know from calculus that for twice continuously differentiable functions $h : U \longrightarrow \mathbb{R}^n$ and $f : V \longrightarrow \mathbb{R}$, with $h(U) \subseteq V$, $U \subseteq \mathbb{R}^r$, for all $v \in U$ we have

$$\nabla^2(f \circ h)(v) = \mathrm{J}\, h(v)^\top \cdot \nabla^2 f(h(v)) \cdot \mathrm{J}\, h(v) + \left(\nabla f(h(v)) \cdot h^*_{ji}(v) \right)_{\substack{j=1,\dots,r \\ i=1,\dots,r}}$$

where $h^*_{ji}(v) = (\partial_j \partial_i h_k(v))_{k=1,\dots,n}$ for each $i, j \in [r]$ is the vector of second-order partial derivatives $\partial_j \partial_i$ of all component functions of h at v, and $\mathrm{J}\, h(v)$ denotes the Jacobian of h at v, and $\nabla^2 f(h(v))$ denotes the Hessian of f at $h(v)$. Applying that to $f \coloneqq \mathsf{SC}$ and $h \coloneqq T$ yields

$$\nabla^2(\mathsf{SC} \circ T)(v) = M_T^\top \cdot \nabla^2 \mathsf{SC}(T(v)) \cdot M_T \ , \tag{2.8}$$

since all second-order partial derivatives of T vanish. The matrix given by (2.8) is also known as the *projected Hessian*. If element latency functions are affine, we have constant Hessian for SC as shown in (2.7), making it easy to check the projected Hessian for being positive semidefinite on $\mathcal{A}'$. We summarize our findings in the following theorem.

2.16 Theorem. *Let all element latency functions be twice continuously differentiable (suffices on $\mathbb{R}_{>0}$). Then* SC *is convex on $\mathcal{A}$ if and only if the projected Hessian* (2.8) *is positive semidefinite for all $v \in \mathcal{A}'$. If all element latency functions are affine, the projected Hessian is independent of v and hence can be checked efficiently*[3] *for being positive semidefinite on $\mathcal{A}'$.*

Finally, if SC is convex on $\mathcal{A}$, optima are characterized by a convex program:

$$\begin{array}{ll} \text{minimize} & \mathsf{SC}(a) \\ \text{subject to} & a \in \mathcal{A} \end{array} \tag{OPT NLP}$$

[3] Up to the numerical inaccuracies involved when, e.g., computing eigenvalues.

2.11.2 Computation of Nash Equilibria

Recall that if $C = R$, Nash equilibria can be characterized by a convex program [11, 18, 19, 37], see also Sec. 1.6. A technically different approach can be taken in the general case, also leading to a characterization of Nash equilibria as optimal solutions to minimization problems, which are not necessarily *convex*, however. Denote $d := (d_1, \ldots, d_N)$ the vector of demands. Then we have:

2.17 Theorem. *A vector $a \in \mathbb{R}^n_{\geq 0}$ is a Nash equilibrium if and only if there exists $\lambda \in \mathbb{R}^N$ such that (a, λ) is an optimal solution to the following NLP:*[4]

$$\begin{array}{ll} \text{minimize} & \mathsf{SC}(a) - \lambda^\top d \\ \text{subject to} & a \in \mathcal{A} \\ & \lambda \in \mathbb{R}^N \\ & \lambda_i \leq L_S(a) \quad \forall S \in \mathfrak{S}_i \ \forall i \in [N] \end{array} \tag{Nash NLP}$$

Proof. The objective function is always non-negative, since for all feasible (a, λ) we have

$$\begin{aligned} \mathsf{SC}(a) &= \sum_{S \in \mathfrak{S}} L_S(a) a_S = \sum_{i \in N} \sum_{S \in \mathfrak{S}_i} \underbrace{L_S(a)}_{\geq \lambda_i} a_S \\ &\geq \sum_{i \in N} \lambda_i \sum_{S \in \mathfrak{S}_i} a_S = \sum_{i \in N} \lambda_i d_i = \lambda^\top d \ . \end{aligned}$$

From this inequality we also immediately deduce that for a feasible (a, λ) we have objective function value 0 if and only if

- $L_S(a) = \lambda_i$ for all $S \in \mathfrak{S}_i$ and all $i \in [N]$ which have $a_S > 0$,
- and, trivially (due to feasibility), $L_S(a) \geq \lambda_i$ for all $S \in \mathfrak{S}_i$ and all $i \in [N]$.

The claim now follows from Prop. 1.5. □

[4] By Thm. 1.6 and Thm. 2.17 we have that the set of solutions to (VAR) coincides with the set of optimal solutions to (Nash NLP). A more general result than this was proved earlier by Aghassi, Bertsimas, and Perakis [1], using stronger tools, including LP duality.

If L_S is concave for all $S \in \mathfrak{S}$ and SC is convex on $\mathcal{A}$, then (Nash NLP) is a convex program. If we have affine element latency functions, then the constraints of (Nash NLP) are linear and the objective function is quadratic. If element latency functions are affine and SC is convex on $\mathcal{A}$, then we essentially have a *linearly constrained convex quadratic* program. We wrote only "essentially", because the objective function of a convex quadratic program is usually specified as:

$$v \mapsto \tfrac{1}{2} v^\top A v + b^\top v, \text{ with } A \text{ being a symmetric, positive semidefinite matrix, and } b \text{ a vector.} \tag{2.9}$$

If $\ell_e(x) = \theta_e x + \tau_e$ with $\theta_e, \tau_e \in \mathbb{R}_{\geq 0}$ for each $e \in E$, and we define matrix $\theta := \operatorname{diag}(\theta_e)_{e \in E} \in \mathbb{R}^{m \times m}$ and vector $\tau := (\tau_e)_{e \in E}$, then

$$\begin{aligned} \mathsf{SC}(a) &= \vec{L}(a)^\top a = (R^\top \vec{\ell}(a))^\top a = (R^\top(\theta \vec{g}(a) + \tau))^\top a \\ &= (R^\top(\theta C a + \tau))^\top a = a^\top (R^\top \theta C)^\top a + (R^\top \tau)^\top a \ . \end{aligned}$$

As we have seen before, the Hessian $\nabla^2 \mathsf{SC} = R^\top \theta C + (R^\top \theta C)^\top$, and so also $R^\top \theta C$, needs not to be positive semidefinite, even if SC is convex on $\mathcal{A}$. However, if SC is convex on $\mathcal{A}$, we can always bring it into the form of (2.9) (plus the terms involving λ) using the affine mapping T. The projected Hessian becomes matrix A, vector b has to be chosen accordingly, and instead of $\mathcal{A}$ we use

$$\{v \in \mathbb{R}^{n-N}_{\geq 0};\ \sum_{S \in \mathfrak{S}'_i} v_S \leq d_i \quad \forall i \in [N]\} \ .$$

Precisely, we have the following linearly constrained quadratic program:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} v^\top M_T^\top \left(R^\top \theta C + (R^\top \theta C)^\top\right) M_T v + (R^\top \tau)^\top M_T v - \lambda^\top d \\ \text{subject to} \quad & v \in \mathbb{R}^{n-N}_{\geq 0} \\ & \sum_{S \in \mathfrak{S}'_i} v_S \leq d_i \quad \forall i \in [N] \\ & \lambda \in \mathbb{R}^N \\ & \lambda_i \leq (R^\top(\theta C M_T v + \tau))_S \quad \forall S \in \mathfrak{S}_i \ \forall i \in [N] \end{aligned}$$

We can take the same approach for optima, receiving a simpler program. As noted earlier, a linearly constrained convex quadratic program is solvable in polynomial time by the ellipsoid algorithm [55] or, up to an arbitrarily small error, by interior-point algorithms [12, 17, 70].

But even in cases where we do not have such a nice method available, the approach via (Nash NLP) is not entirely hopeless; a vector returned by some algorithm (e.g., a solver for general non-linear programs) can be checked easily for being a Nash equilibrium by considering the corresponding objective value of (Nash NLP) or by checking directly whether the solution fulfills the definition of Nash equilibrium. So we can, at least, always determine whether some proposed solution is in fact correct.

2.11.3 Extreme Nash Equilibria

Nash equilibria need not to be unique, and two different equilibria may have very different social cost, as we have seen before in Sec. 2.2. We add, for some numbers $c_0 \leq c_1$, the additional linear constraint $c_0 \leq \lambda^\top d \leq c_1$ resulting in the following non-linear program.

$$\begin{aligned} \text{minimize} \quad & \mathsf{SC}(a) - \lambda^\top d \\ \text{subject to} \quad & a \in \mathcal{A} \\ & \lambda \in \mathbb{R}^N \\ & \lambda_i \leq L_S(a) \quad \forall S \in \mathfrak{S}_i \ \forall i \in [N] \\ & c_0 \leq \lambda^\top d \leq c_1 \end{aligned} \tag{Nash NLP'}$$

If we can check feasibility of (Nash NLP'), we can check whether there exists a Nash equilibrium with social cost between c_0 and c_1. If we can, in case of feasibility, solve (Nash NLP'), we can find a Nash equilibrium with social cost between c_0 and c_1. For affine element latency functions and SC convex on $\mathcal{A}$, this is both possible: the constraints are linear then and the objective function is convex quadratic.

Let $\tilde{\rho}$ denote an upper bound on the price of anarchy, and assume that we know OPT. Then we know that all Nash equilibria have social cost in the interval $[\mathsf{OPT}, \tilde{\rho}\,\mathsf{OPT}]$. Using binary search, we can hence

compute a worst or a best equilibrium with an error of ε by solving at most $\lceil \log_2 \frac{\tilde{\rho}-1}{\varepsilon} \rceil$ instances of (Nash NLP′), provided the latter is practical. We summarize our findings in the following theorem. A detailed description and analysis of a binary search algorithm will be given in the next chapter.

2.18 Theorem.

(i) Optima and Nash equilibria are characterized by the non-linear programs (OPT NLP) *and* (Nash NLP)*, respectively.*

(ii) Let $c_0 \leq c_1$. If (Nash NLP′) *is feasible and has optimal value* 0*, it characterizes all Nash equilibria with social cost between c_0 and c_1. If* (Nash NLP′) *is infeasible or has optimal value > 0, no Nash equilibrium with social cost between c_0 and c_1 exists.*

(iii) Let $\tilde{\rho}$ be an upper bound on the price of anarchy. We can compute[5] *a worst or best Nash equilibrium up to an error of ε (and possibly an additional error introduced when solving the involved non-linear programs) by solving one instance of* (OPT NLP)*, and by solving or showing to be infeasible at most $\lceil \log_2 \frac{\tilde{\rho}-1}{\varepsilon} \rceil$ instances of* (Nash NLP′)*.*

(iv) If L_S is concave for all $S \in \mathfrak{S}$ and SC *is convex on $\mathcal{A}$, then both* (OPT NLP) *and* (Nash NLP′) *are* convex *programs.*

(v) If element latency functions are affine and SC *is convex on $\mathcal{A}$, then both* (OPT NLP) *and* (Nash NLP′) *are* linearly constrained convex quadratic *programs.*

In total, we have a satisfactory computational result for affine element latency functions: by Thm. 2.16 we can check whether SC is convex on $\mathcal{A}$ or not, and if so, we can compute optima and extreme Nash equilibria (up to a small error) by convex quadratic programming.

Practicability may, however, depend on the number of variables – and so for our programs in particular on the number of strategies. If, e.g., the set of strategies was to enumerate *all* possible multicast path

[5] This statement is to be understood relative to the practicability of solving (OPT NLP) and (Nash NLP′). It is not intended to make any claim that we actually can solve these programs, it merely states the number of different instances that would be needed to be solved. See also statements (iv) and (v).

sets (or trees) in a network, this could be exponential in the number of links. Efficiently handling such applications is left for future work.

2.12 Bibliographic Remarks

The price of anarchy in selfish multicast routing or a model similar to the NCRCG model has to the best of my knowledge not been studied before – except for [9], where we published early results on selfish multicast routing, and except for approaches which cover the NCRCG model partly, e.g., the work by Perakis [72].

Boulogne and Altman [16] in 2005 studied existence and uniqueness issues in selfish multicast routing in an atomic and a non-atomic model. They considered duplication flows and also discern edge-based and path-based latency, see definitions "(A)" and "$(B.1)$" in [16, p. 24]. Regarding the non-atomic model, it was claimed that all Nash equilibria induced the same congestions under path-based latency, and so are essentially unique. This is disproved by our example in Sec. 2.2. The flaw lies in an incorrect re-arrangement of sums in the proof of [16, Lem. 4.3].

2.13 Summary

We studied the price of anarchy in NCRCGs. NCRCGs are a generalization of NCGs, motivated by selfish multicast routing. We introduced a new parameter γ and showed how we can lower- and upper-bound the price of anarchy using γ and the known parameter β, and, in case of polynomial element latency functions, also the maximum degree p. For polynomial element latency functions from $\text{Poly}^+(p)$, we proved an upper bound on the price of anarchy of:

$$\begin{cases} \gamma^{p+1} & \text{if } \gamma \geq (1+p)^{\frac{1}{p}} \\ \frac{1}{1-\gamma\beta}\gamma & \text{if } \gamma \leq (1+p)^{\frac{1}{p}} \end{cases},$$

where $\beta = \beta(\text{Poly}^+(p)) = p\,(p+1)^{-1-\frac{1}{p}}$. For comparison, the known tight bound for NCGs is just $\frac{1}{1-\beta}$. We then proved a lower bound, being

only a factor γ away from our upper bound. The upper bound also comes in a more general version, namely for s-super-homogeneous element latency functions.

Then we considered bicriteria bounds. We proved that when scaling up demands by $(1 + \beta(\mathcal{L}))\,\gamma$, even an optimum is no less costly than a Nash equilibrium for the original demands. This provides a natural extension to previous results on NCGs, where a scaling factor of $1 + \beta(\mathcal{L})$ occurs. We also showed that for polynomials, the new scaling factor is the best possible up to a factor of $(1 + \beta(\mathcal{L}))\,\gamma^{\frac{1}{1+p}}$.

Finally, we considered computation of optima and (extreme) Nash equilibria. We showed how both can be characterized by non-linear programs and when and how convexity of the objective function can be determined. The objective function is essentially the social cost function SC with an additional linear term. We pointed out cases where we have linearly constrained convex quadratic programs and also described a heuristic procedure for the general case. We will show how to put that into practice in the next chapter.

Chapter 3

Experimental Studies and Conjecture

This chapter contains the first thorough experimental investigation of the price of anarchy in NCRCGs with polynomial element latency functions from $\text{Poly}^{+}(p)$ for $p \in \{1, 2, 3\}$ and one player class. The driving force for the experiments are the following three questions:

- Are there instances that provide a higher lower bound than given in Thm. 2.9?
- How does a binary search scheme based on Thm. 2.18 perform in practice?
- How do our upper bound on the price of anarchy (Thm. 2.12) and Perakis' bound (Thm. 1.25) compare, in cases where both are applicable?

Our studies are based on several million randomly generated small instances. We describe our computational procedure in Sec. 3.1, the experimental setup in Sec. 3.2, and the random model in Sec. 3.3. Results are presented and discussed in Sec. 3.4 and Sec. 3.5. The main result is that the lower bound of Thm. 2.9 essentially remains the best one known. We did not succeed in finding an example for which we could show a price of anarchy exceeding the bound by more than 1% for $p \in \{1, 2\}$ and 4% for $p = 3$. We attribute these small overstepping to numerical inaccuracies. Moreover, for many settings of parameters, we could span the whole range between 1 and the bound with examples.

This is an indication that neither our computational procedure nor the way we generate random instances is responsible for us not observing higher prices of anarchy. We hence conjecture that this bound, namely the lower bound from Thm. 2.9, is in fact an upper bound (for polynomials).

Conjecture. *The price of anarchy in an NCRCG with element latency functions drawn from* $\mathrm{Poly}^+(p)$ *is no more than*

$$\begin{cases} \gamma^p & \text{if } \gamma \geq (1+p)^{\frac{1}{p}} \\ \frac{1}{1-\gamma\beta} & \text{if } \gamma \leq (1+p)^{\frac{1}{p}} \end{cases}, \quad \text{where } \beta = \beta(\mathrm{Poly}^+(p)) = p\,(p+1)^{-1-\frac{1}{p}}.$$

It is worth noting that the finer facets of this conjecture were obtained *during* the experiments. The starting point was a conjecture inspired by the universal bound from Thm. 2.12 directly applied to polynomial element latency functions, namely we conjectured an upper bound of simply $\frac{1}{1-\beta}\gamma^p$. Most of the subsequent refinements that lead to Thm. 2.9, Thm. 2.12, and finally to the above conjecture were inspired by experimental results. We depict this more closely on p. 177 and p. 178.

Additionally, in Sec. 3.7, we compare our bounds with Perakis' where applicable. We give a summary in Sec. 3.8. Experimental results are given in detail in App. B; these plots and tables are explained in this chapter in Sec. 3.5 to Sec. 3.7.

3.1 Computational Procedure

For each randomly generated instance, we do the following:

1.) If element latency functions are affine, then check whether SC is convex on $\mathcal{A}$ or not. This is done by checking whether the projected Hessian is positive semidefinite.

2.) If element latency functions are affine, then check whether Perakis' bound for the positive definite case is applicable. If so, then compute it.

Algorithm 2: Binary search for a worst Nash equilibrium.

```
left ← OPT;
right ← ρ̃ OPT;
I ← ⌈log_2 1/(ε OPT) (right − left)⌉  (= ⌈log_2 (ρ̃−1)/ε⌉);
i ← 0;
c_0 ← left;
while i < I do
    if (Nash NLP) with c_0 is feasible then
        ã ← optimal solution to (Nash NLP) using c_0;
    else
        ã ← NIL;
    if ã is acceptable then
        a ← ã;
        left ← c_0;
    else
        right ← c_0;
    c_0 ← left + (right−left)/2  (= (left+right)/2);
    i ← i + 1;
return a;
```

3.) Try to find an optimum a^* of (OPT NLP), using an NLP solver back-end [31, 86, 89].

4.) Try to solve (Nash NLP) using the NLP solver back-end with different c_0, following a binary search scheme. The Nash equilibrium a with the highest social cost obtained by this procedure is returned as the result.

We describe the last step in more detail in the following. Recall the non-linear program for Nash equilibria from Sec. 2.11.

$$\begin{aligned} \text{minimize} \quad & \mathsf{SC}(a) - \lambda^\top d \\ \text{subject to} \quad & a \in \mathcal{A} \\ & \lambda \in \mathbb{R}^N \\ & \lambda_i \leq L_S(a) \quad \forall S \in \mathfrak{S}_i \ \forall i \in [N] \\ & c_0 \leq \lambda^\top d \end{aligned} \tag{Nash NLP}$$

For now, we call a vector $a \in \mathbb{R}^n$ *acceptable* when it is a Nash equilibrium; this will be refined for practical purposes later in Sec. 3.2.

3.1 Theorem. *Let $\tilde{\rho}$ be an upper bound on the price of anarchy ρ. Assuming that we can determine the feasibility of* (Nash NLP) *and solve it optimally if it is feasible, Alg. 2 on the previous page computes the price of anarchy up to a relative accuracy of ε, i.e., $\rho - \frac{\mathsf{SC}(a)}{\mathsf{OPT}} \leq \varepsilon$ for the solution a.*

Proof. All Nash equilibria have their social cost within the interval $[\mathsf{OPT}, \tilde{\rho}\,\mathsf{OPT}]$. This algorithm always finds a solution, since there is a Nash equilibrium of social cost at least OPT. Let a_{w} be a worst Nash equilibrium, so the price of anarchy is $\rho = \frac{\mathsf{SC}(a_{\mathrm{w}})}{\mathsf{OPT}}$. Then by the resolution of the binary search, the result a of the algorithm fulfills $\mathsf{SC}(a_{\mathrm{w}}) - \mathsf{SC}(a) \leq \varepsilon\mathsf{OPT}$ and so the error is $\rho - \frac{\mathsf{SC}(a)}{\mathsf{OPT}} \leq \varepsilon$. □

3.2 Definition. By *observed price of anarchy* we refer to the value $\frac{\mathsf{SC}(a)}{\mathsf{SC}(a^*)}$, where a^* and a are as in 3.) and 4.) on page 76, respectively.

If we can correctly determine feasibility of the involved NLPs and solve them optimally if feasible, then the observed price of anarchy equals the price of anarchy, up to the error ε introduced by binary search. In general, the observed price of anarchy is a lower bound on the price of anarchy.

To better quantify how close the observed price of anarchy is to the bound of the conjecture, we introduce the *target ratio* of an instance. It is the ratio of observed price of anarchy to conjectured bound. A target ratio of 1 means that the conjectured bound is hit, and a target ratio of more than 1 would disprove the conjecture. We say that an instance has TRx if its target ratio is x% or more, e.g., an instance with TR90 has a target ratio of at least 0.9. Since the conjecture is known to be true for $\gamma = 1$, we are mostly interested in the target ratios of instances with $\gamma > 1$. Later we will give numbers stating what percentage of the randomly generated instances with $\gamma > 1$ have a certain TRx (we call this percentage "TRx value"), and we will also state how many instances have $\gamma > 1$.

For some instances in our experiments, the binary search could not find *any* acceptable solution. This happens rarely for $p = 1$ and

more frequently than that for $p > 1$, as can be seen in the tables found in App. B.2 starting on p. 179, in the column named "%OK". If it happens, the most common cause is that the vector returned by the NLP solver fails the test for being a Nash equilibrium. As a possible explanation, it was considered that the solver was not adjusted to produce sufficiently accurate solutions. Additional tests with the NLP solver configured for higher accuracy, however, suggested that this is not likely to be the cause. Another explanation, for the cases where only the Nash condition is not met, is that the solver simply failed to find a global optimum. This is substantiated by the observation that no instance with SC convex on $\mathcal{A}$ showed that kind of problem (for $p = 1$).

3.2 Experiment Setup

We wrote an implementation in C++, using various libraries. Most importantly, we use Ipopt [86, 89] (version 3.7.1) to solve the non-linear programs involved in the computation of Nash equilibria. We use a combination of Ipopt and Lancelot [31] to solve the non-linear programs involved in the computation of optima. For eigenvalue computations, needed for Perakis' bound, we use the GNU Scientific Library [44]. Post-processing and plotting of gained data is done using the R System [74]. Experiments are run in parallel on several Barcelona® and Opteron® multicore machines using the Linux operating system at the Rechenzentrum Universität Kiel.

We apply the following counter-measures and workarounds for numerical inaccuracies:

– As noted earlier, all γ values are with respect to an optimal scaling, in order to reduce the bounds. That is, we use

$$\gamma_{\text{scaled}} := \min_{\varepsilon_1,\ldots,\varepsilon_m>0} \left\{ \max_{e\in E,S\in\mathfrak{S}} \frac{C_{eS}}{\varepsilon_e R_{eS}} \cdot \max_{e\in E,S\in\mathfrak{S}} \frac{\varepsilon_e R_{eS}}{C_{eS}} \right\} .$$

  We do, however, not actually scale the instances to not risk numerical instabilities. Since a scaled instance has the same optima and Nash equilibria as the original one, this procedure is correct.

– Instead of checking $\gamma > 1$ for determining the target ratios, we check for $\gamma > 1.01$.

– We initialize the binary search with $\mathit{left} \leftarrow 0.95\,\mathsf{OPT}$. The accuracy used in the binary search is $\varepsilon = 10^{-4}$.

– A vector $a \in \mathbb{R}^n$ is considered *acceptable* if all of the following conditions hold:

  (i) the NLP solver does not signal an error,

  (ii) $|d - \sum_{S\in\mathfrak{S}} a_S| \le 10^{-2} d$ and $a_S \ge 0$ for all $S \in \mathfrak{S}$,

  (iii) $c_0 - \mathsf{SC}(a) \le c_0 10^{-2}$,

  (iv) $a_S(L_S(a) - \Lambda(a))) \le d\Lambda(a)10^{-3}$ for all $S \in \mathfrak{S}$, where $\Lambda(a) := \min_{S\in\mathfrak{S}} L_S(a)$.

  We generally rely on the NLP solver to signal errors and infeasibility correctly, so (ii) and (iii) exist mostly as a precaution. In fact no instance failed (ii) or (iii) without failing (i). Test (iv) is to determine whether we have a Nash equilibrium or not.

– We use the following settings for Ipopt:

```
tol 1e-10
dual_inf_tol 1e-02
constr_viol_tol 1e-06
compl_inf_tol 1e-06
acceptable_tol 1e-08
acceptable_dual_inf_tol 1e+08
acceptable_constr_viol_tol 1e-04
acceptable_compl_inf_tol 1e-04
max_iter 5000
```

  With these settings combined with those of the previous item, we obtain accurate solutions when rounded to two decimal digits for a set of known instances.

– Tests for matrix definiteness are done comparing eigenvalues not against 0, but against 10^{-13}. This had turned out to be necessary in order to prevent problems due to numerical errors, e.g., the Hessian being recognized as positive semidefinite but the projected Hessian not, or Cholesky decomposition (needed for Perakis' bound) failing on an allegedly positive definite matrix.

3.3 Random Model and Data Sets

Since the NCRCG model comprises quite a few different parameters, which have several constraints to fulfill, it is not directly obvious how to generate random instances. We chose the following way. Recall that we restrict to one player class.

1.) Fix the number of elements m, number of strategies n, the demand d, and a maximal degree p.

2.) For each $k \in \{0, \dots, p\}$ fix $0 \leq \theta_{k,\min} \leq \theta_{k,\max}$ and $P_k \in [0, 1]$.

3.) Fix numbers $0 < c_{\min} \leq c_{\max}$ and $0 < r_{\min} \leq r_{\max}$.

4.) For each strategy S, fix $P_{S,\min}, P_{S,\max} \in [0.0, 1.0]$.

5.) For each strategy S, choose a number $P_S \in [P_{S,\min}, P_{S,\max}]$ uniformly at random.

6.) For each strategy $S \in \mathfrak{S}$:

Initialize $P \coloneqq P_S$. Then for $e = 1, \dots, m$ do: with probability $1 - P$ set C_{eS} and R_{eS} to 0, and with probability P choose C_{eS} from $[c_{\min}, c_{\max}]$ uniformly at random and R_{eS} from $[r_{\min}, r_{\max}]$. Until the strategy has at least one non-zero entry, increase P after each step in such a way that if $e = m$ and still no non-zero entry exists, we will have $P = 1$. After a non-zero entry was chosen, reset $P \coloneqq P_S$ and never touch it again until the next strategy is treated. This is just one way to ensure our requirement that there must be no empty strategy.

7.) For each element $e \in E$:

For each $k \in [p]$ with probability $1 - P_k$ set $\theta_{e,k}$ to 0, and with probability P_k choose $\theta_{e,k}$ from $[\theta_{k,\min}, \theta_{k,\max}]$ uniformly at random. If all so chosen coefficients are zero, start over again for that element. Finally define $\ell_e(x) \coloneqq \sum_{k=0}^{p} \theta_{e,k} x^k$. This procedure ensures that $\ell_e(x) > 0$ for all $e \in E$ and $x > 0$, and so that $\mathsf{OPT} > 0$.

We do experiments in series. For each series, steps 1.) to 4.) are executed once, and then steps 5.) to 7.) are executed many times, and computations are done for each instance. We now describe the scheme by which we fix the parameters in steps 1.) to 4.). We fix the following parameters for all experiments: $m := 4$ and $[P_{S,\min}, P_{S,\max}] := [0.1, 1.0]$ for all $S \in \mathfrak{S}$. We also fix $c_{\min} := r_{\min} := 1$, which is without loss of generality by a scaling argument.

The number of strategies n ranges in $\{2, \ldots, 6, 9\}$, which was chosen to cover the range where n is not far from m, and also to show the effects of n being larger than m. Regarding demand, we run each experiment once with $d := 1$ and once with $d := 10$, in order to cover two different scales of congestions, latencies, and social costs.

Latency Functions. The maximum degree of polynomials in element latency functions p ranges in $\{1, 2, 3\}$. Fix one p for now. We use two different sets of parameters for the element latency functions. Instances using the first set of parameters are called *single*: $P_p := 1.0$ and $P_k := 0$ for $k \neq p$, $[\theta_{k,\min}, \theta_{k,\max}] := [1.0, 1.0]$ for all k. That is, we have $\ell_e(x) = x^p$ for all $e \in E$. We denote such a vector of parameters by $single := (\theta_{k,\min}, \theta_{k,\max}, P_k)_{k \in \{0,\ldots,p\}}$.

Instances using the second set of parameters are called *mixed*: we set $P_k := 0.5$ and $[\theta_{k,\min}, \theta_{k,\max}] := [0.01, 1.0]$ for all $k > 0$, and $P_0 := 0.5$ and $[\theta_{0,\min}, \theta_{0,\max}] := [0.01, c_{\max} r_{\max}]$. That is, for each element latency function, each of the coefficients has a probability of 0.5 to be non-zero, and if so, it is chosen uniformly at random from the interval $[0.01, 1.0]$ in case of a non-constant term and $[0.01, c_{\max} r_{\max}]$ for the constant term. The 0.01 is to prevent too small numbers and the $c_{\max} r_{\max}$ is inspired (for $p = 1$) by the example in the proof of Thm. 2.9(ii). This will lead to a variety of element latency functions: constant ones and non-constant ones, and ones with or without constant term. We denote such a vector of parameters by $mixed := (\theta_{k,\min}, \theta_{k,\max}, P_k)_{k \in \{0,\ldots,p\}}$.

Consumption and Relevance Numbers. For one group of series, we fix the range for consumption numbers to $[1.0, 9.0]$ and then go through three different intervals for the relevance numbers: $[1.0, 1.0]$, $[1.0, 5.0]$, and $[1.0, 9.0]$. For the other group of series, we fix the range for relevance numbers to $[1.0, 9.0]$ and then go through three different intervals for the consumption numbers, just as above for relevance numbers.

Table 3.1. Running times for increasing numbers of strategies. Each number gives the total running time in seconds for $192 \cdot N_p = 19{,}200$ instances with the specified parameters on a 2.7 GHz machine, divided by 19.2 in order that it corresponds to 1,000 instances. These are only rough measurements: generation of instances, scaling, computation of bounds, IO, etc. is all included.

n	2	3	4	5	6	7	8	9
$p = 1$	76	92	98	106	127	154	196	246
$p = 2$	90	118	135	157	199	233	304	389
$p = 3$	177	203	194	253	324	398	480	664

Summary. We run the following experiments. Fix $m := 4$. For each $p \in \{1, 2, 3\}$, each $d \in \{1, 10\}$, each

$$(\theta_{k,\min}, \theta_{k,\max}, P_k)_{k \in \{0,\dots,p\}} \in \{single, mixed\} \quad ,$$

and each $n \in \{2, 3, 4, 5, 6, 9\}$ do:

1. (Fixed consumption number range.) Fix $[c_{\min}, c_{\max}] := [1.0, 9.0]$ and $r_{\min} := 1.0$. For each $r_{\max} \in \{1.0, 5.0, 9.0\}$ generate N_p random instances and compute their observed price of anarchy, bounds, and parameters.
2. (Fixed relevance number range.) Fix $[r_{\min}, r_{\max}] := [1.0, 9.0]$ and $c_{\min} := 1.0$. For each $c_{\max} \in \{1.0, 5.0, 9.0\}$ generate N_p random instances and compute their observed price of anarchy, bounds, and parameters.

The numbers N_p are $N_1 := 32{,}000$, $N_2 := 32{,}000$, and $N_3 := 16{,}000$. We treat fewer instances of $p = 3$ due to their higher computational overhead. Tab. 3.1 gives a brief impression of running times. These measurements were taken for one run of all experiments as described above, with $N_p = 100$ and itemized by maximum degree p and number of strategies n. We additionally included the cases $n \in \{7, 8\}$ here.

3.4 Qualitative Observations

We first state some qualitative observations and in the next sections substantiate them by quantitative results. All statements concerning convexity and Perakis' bound are only based on the data gained for $p = 1$.

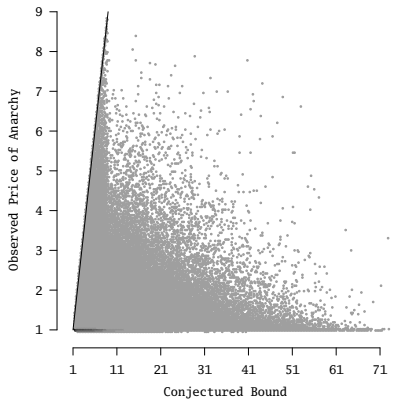
– Provoking an observed price of anarchy that is close to the conjectured bound requires the choice of specific parameters. For many settings, the observed price of anarchy usually is far away from that bound, as can be seen by the TRx values, especially for $n \geq m$.
– We observed the more instances with an observed price of anarchy close or on the conjectured bound the smaller $r_{\max} - r_{\min}$ was compared to $c_{\max} - c_{\min}$. This can be seen by comparing the TR90 values from series with fixed consumption number range, which is $[1.0, 9.0]$, and small relevance number range against values from series with fixed relevance number range.
– For fixed m, the fraction of instances with SC convex on $\mathcal{A}$ decreases with increasing n, and that decrease is rapid once $n \geq m$.
– It makes a substantial difference using the projected Hessian instead of the Hessian itself for determining the convexity of SC on $\mathcal{A}$. There are many instances with SC convex on $\mathcal{A}$ but not globally convex, i.e., not convex on $\mathbb{R}^n$.

  A globally convex SC is equivalent to $\mathrm{J}\,\vec{L}$ being positive semidefinite, and hence is equivalent to one of Perakis' bounds being applicable. Since so many instances have no globally convex SC, Perakis' bounds are also not applicable for many instances.
– We found only a very few instances where Perakis' semidefinite bound would have been applicable but the positive definite bound not; see the differences between columns number four and five of each table in App. B.2.1.

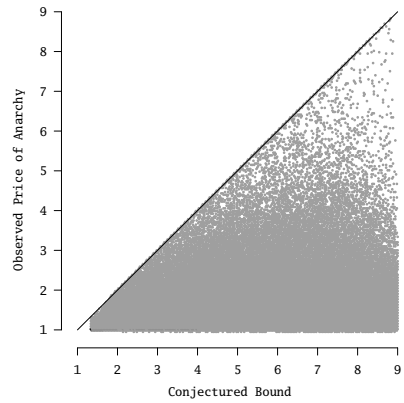
3.5 Hexagonal Binning Plots

We use kinds of scatter plots – namely hexagonal binning plots – to receive an impression of the distribution of observed price of anarchy in relation to the conjectured bound. A scatter plot is a method to visualize two-dimensional data, e.g., a set of pairs of the form

$$(\text{conjectured bound, observed price of anarchy}) \ .$$



a. Random instances with $p = 1$ from all the series with fixed consumption number range.



b. Random instances with $p = 1$ from all series. Only those are shown which have conjectured bound at most the maximum observed price of anarchy.

Figure 3.1. Hexagonal binning plots. Larger versions of these and other plots are given in App. B.

For each such pair, a point is drawn in the plane at the coordinates given by the pair. In a scatter plot for our application, a dot is drawn for every single instance. The horizontal position of the dot is determined by the conjectured bound, and the vertical position is determined by the observed price of anarchy.

A hexagonal binning plot is similar to a scatter plot, and is better suited for large data sets like the ones we have. For a hexagonal binning plot, the plane is tessellated with small hexagons. Then an imaginary scatter plot is conducted, and for each hexagon it is counted by how many points it is hit. Finally, each hexagon is filled with a shade of gray corresponding to the number of hits. We use the `hexbin` R package [24] to create hexagonal binning plots. The number of hexagons is chosen large, in order to have a fine resolution; the plots thus closely resemble the underlying scatter plots. Darker shades of gray indicate a higher number of hits, i.e., a higher density of points. Our plots only show a higher density in the lower left corner, if at all.

Our plots are shown in App. B.1 starting on p. 172; two small examples are also given in Fig. 3.1 on the previous page. In addition to hexagons, a black line, which we call the *borderline*, is drawn, that marks where the observed price of anarchy equals the conjectured bound, i.e., it is simply a plot of the function $x \mapsto x$. Instances represented by hexagons close to that line have TRx for x close to 100, and a hexagon positioned to the upper left of that line would have indicated a counter-example to the conjecture. A few are positioned vertically below 1, which is an impossible value for the price of anarchy. A possible explanation is again numerical inaccuracies or that the NLP solver failed to solve (OPT NLP) optimally. These dropout cases disappear when we remove the non-convex cases from the data set.

3.6 Tables

Our results are shown in detail in several tables in App. B.2. The eight tables starting with Tab. B.1 on p. 179 show results for $p = 1$ and several parameters with 32,000 randomly generated instances in each row. Numbers are rounded to the shown number of decimal digits. The first column gives the number of strategies. The second column gives the upper bound on the range for consumption numbers or relevance numbers, respectively. The next two columns give the percentage of instances with SC convex on $\mathcal{A}$ (abbreviated "C") and globally convex SC (abbreviated "GC"), respectively. The next column shows the percentage of instances for which Perakis' positive definite bound is applicable.

The sixth column (titled "%OK") gives the percentage of instances where the binary search found at least one acceptable solution. Only these instances form the basis for the values in the following columns. The next column gives the average price of anarchy. The eighth column (titled "#($\gamma > 1$)") gives the number of instances with $\gamma > 1$ (recall that we in fact test for $\gamma > 1.01$). In the following three columns we have the percentages of instances with the specified TRx property (relative to the number of instances with $\gamma > 1$) with $x \in \{50, 90, 100\}$. The reason we give column number eight is to show how many instances count for the TRx statistics.

Starting with Tab. B.9 on p. 187 we give more TRx values, namely for $x \in \{30, 40, \ldots, 100\}$ and for $p \in \{1, 2, 3\}$.

3.7 Comparison with Perakis' Bound

We compare Perakis' bound for the positive definite case to our proven and our conjectured bound. Results are shown in App. B.3, starting on p. 212. The basis are the 4,587,662 instances with $p = 1$ for which the binary search yielded at least one acceptable solution. From those, we pick those in which Perakis' bound is applicable. This is approximately 20% of the instances, namely 926,254 instances in total. On p. 212 we show a comparison with our proven bound by the means of two plots. Perakis' bound looks better than ours. However, the computation of Perakis' bound also yielded some extraordinarily high values, up to the value of 9,007, which were removed from the data set before creating the plot for the sake of a reasonable scale. It is currently not understood whether these high values are to be attributed to numerical errors, or reflect the true values. On p. 213 and p. 214 a comparison with our conjectured bound is given. Based on this plot we recognize a good performance of ours compared to Perakis' bound. The values of our bound barely reach beyond 50, whereas the values of Perakis' bound span the whole horizontal scale. Moreover, 1,395 values of Perakis' bound are off-scale.

For a more quantitative evaluation, we give two histograms on p. 215. They show the distribution of the ratios of Perakis' bound to our proven and conjectured bound, respectively. A ratio less than 1 means that Perakis' bound is smaller (hence better), and a ratio of more than 1 means that our bound is better. Relative to this measure, we recognize by looking at the histograms that both our bounds are outperformed by Perakis' most of the time.

We conclude that our bounds have the following strengths:

- They are always applicable, not only under Jacobian definiteness conditions.
- They are computationally simple to obtain.
- They can easily be roughly upper-bounded if we know ranges of

consumption and relevance numbers. This rough bound is $9 \cdot 9 = 81$ in our experiments for our conjectured bound.

Perakis' bound – if applicable – has the strength of being smaller than ours, even our conjectured bound, in most cases.

3.8 Summary

We presented an experimental framework for the NCRCG model and put the theoretical computational results of Sec. 2.11 into practice. The per-instance running time for our small instances is close to negligible. Expectedly, it grows with the number of strategies and the degree of the polynomials. Regarding the quality of the solution delivered, it was pleasant to see that we found at least one Nash equilibrium per instance for the majority of instances (cf. column "%OK" in the tables in App. B.2). We computed lower bounds on the price of anarchy – the "observed price of anarchy" – up to the best known theoretical lower bound, but not exceeding it (up to inaccuracies). The observed prices of anarchy build up a line along the borderline drawn by the theoretical lower bound, as depicted by the plots in App. B.1. Based on this, we conjectured that the lower bound is also an upper bound.

We also compared our bounds to Perakis' bound and pointed out strengths and weaknesses of both approaches.

Conclusion and Future Directions

We have witnessed how a seemingly small change to the model of non-atomic congestion games (NCG) affects profound structural and quantitative changes, and qualitative new phenomena arise. All we did was to exchange the rates of consumption for two sets of numbers: consumption numbers and relevance numbers. This is the non-atomic consumption-relevance congestion game (NCRCG) model. The model extension was motivated by the extension of selfish unicast to multicast routing, but of course it is also interesting in its own right. The most important changes are:

– The price of anarchy in NCRCGs, even from a worst-case point of view, depends on structural parameters not limited to element latency functions (Sec. 2.2 and Sec. 2.8). As the relevant parameter, we introduced γ (Sec. 2.5). If the NCRCG models multicast, this parameter is related to the network structure. For comparison: in NCGs, the worst-case price of anarchy only depends on the latency functions used (Thm. 1.16, Thm. 1.18, and Thm. 1.17).

– If considering polynomial element latency functions, the worst-case price of anarchy in NCRCGs depends exponentially on the maximum degree p. (Sec. 2.2 and Sec. 2.8). For comparison: in NCGs, we have a $\Theta(\frac{p}{\ln p})$ bound. So roughly speaking, NCRCGs tend to exhibit a much higher price of anarchy than NCGs.

– NCRCGs admit Nash equilibria with different social cost. For polynomials from $\mathrm{Poly}^{+}(p)$, there may even be a gap exponential in p between the social costs of different equilibria (Sec. 2.2). For comparison: in an NCG, all Nash equilibria have the same social cost (Sec. 1.5).

We see the following challenges for future work:

1. Prove or disprove the conjectured upper bound on the price of anarchy.
2. Study the *price of stability*, i.e., $\inf_{a \text{ is N.E.}} \frac{\mathsf{SC}(a)}{\mathsf{OPT}}$. This is not an issue in the NCG model, since all Nash equilibria have the same social cost in an NCG and hence price of stability and price of anarchy coincide. For NCRCGs, as seen in Sec. 2.2, the gap between price of anarchy and price of stability can be exponential in p for element latency functions from $\text{Poly}^+(p)$. It appears that the basic tool, the variational inequality (VAR) on p. 14, which characterizes Nash equilibria, is insufficient for bounding the price of stability. New methods will be required, which hopefully might also yield a proof for the conjectured bound on the price of anarchy.
3. Develop new algorithms to compute optima and Nash equilibria. Our experimental results in Ch. 3 and App. B.2 can serve as a benchmark for future experimental studies: will a new algorithm be able to deliver substantially higher TRx values for the same random model? Will a new algorithm succeed in disproving the conjecture by the discovery of a counter-example?
4. Analyze the causalities between the random model from Sec. 3.3 and the observations and results in Sec. 3.4.
5. Address the computational challenge in case that the number of strategies is too large in order to store C or R explicitly in memory or to treat mathematical programs that contain one variable for each strategy. In unicast routing, if all paths are eligible for routing, or if the exclusion of certain paths can be expressed by exclusion of edges, everything of interest can be expressed on edge level using flow conservation constraints, as described in Sec. 1.6. A representation as an action distribution can be reconstructed from this, if desired. It is unclear how to do something similar for general NCRCGs.

Part II

Distributed Network Formation

Chapter 4

Distributed Network Formation

In Part I a network was given and a large number of players used it for routing in a distributed manner. We saw that a network-related parameter γ influenced the price of anarchy. Now, we consider a network being *built* in a distributed manner. Many details will be different, however, there are also similarities:

- Players wish to connect to each other, so there is a routing aspect.
- We will use the concept of Nash equilibrium (besides another equilibrium concepts).
- In Ch. 5, we will again meet the basic notion "how many terminals are reached via this edge" as a central concept.

Still, this part can be understood without the knowledge of Part I.

We mostly review previous work in this chapter and introduce the relevant modeling and equilibrium concepts. New results will be presented in Ch. 5.

In Sec. 4.1 and Sec. 4.2, we explain the basics of the model, including the different ways that links can be formed (namely unilaterally or bilaterally), individual and social cost functions, and Nash equilibrium. Nash equilibrium is not the only equilibrium concept we will use, but we restrict to it at first. Equipped with all necessary prerequisites, we study a concrete model in Sec. 4.3, the *sum-distance model*. In Sec. 4.4 we look at the bilateral case more closely and discuss several alternative equilibrium concepts. We show that the sum-distance model behaves differently under bilateral link formation. We give a bibliographic overview in Sec. 4.5.

We only use graphs and directed graphs in this part, we do not need multigraphs or directed multigraphs. For basic graph-theoretic terminology we refer to App. A.2.

4.1 Basic Idea

We work in an *atomic* model, that is, there is a finite number of players, each having a substantial influence on the system. Each player represents a vertex in the to-be-built network. We use the terms *player* and *vertex* synonymously. Let n be the number of players, and denote $V = [n] = \{1, \ldots, n\}$. So, while "$n$" denoted the number of strategies in the previous part, now it is the number of players. Generally, the models in this part have fewer parameters than those in the previous part; in particular, we do not have to consider latency functions.

Players can build edges to other players, hence forming a graph. Building edges incurs a cost – however, a cost is also incurred when the graph has bad routing properties. We describe the basic idea informally by a dynamic process, although we will only do static analysis later. Consider n vertices without any edges. Each vertex wishes to be able to route to each other vertex, and moreover the connectivity should be 'good' in some sense, e.g., with short path lengths. The empty graph is clearly not suitable. So players will start building edges. Each player v can build edges of the form $\{v, w\}$. All edges can be used in both directions for routing. Each edge that player v builds, costs an amount of α, where $\alpha > 0$ is a parameter. Hence, depending on how great α is, it may not be desirable for a player to build an edge to every other vertex, although this would probably give a good connectivity. The connectivity is expressed by an additional cost, which we call the *indirect cost*. Good connectivity means small, or even zero indirect cost. Bad connectivity means high indirect cost. For instance, the indirect cost for player v could be the sum of distances to all other players. The individual cost of each player is the cost incurred by building edges, the *building cost*, plus the indirect cost. The social cost is the sum over the individual costs of all players.

Several questions arise. What do equilibrium graphs look like, that is, graphs in which, say, no player has an incentive to build any

additional edges, remove existing edges, or exchange some currently built edges for others?[1] How high is the social cost in equilibrium graphs? How much does it differ from the cost of an optimum graph, i.e., what is the price of anarchy?

4.2 Model Framework

We introduce the model framework and all related notation here, that will be used throughout this part. This is only a framework, not a complete model; the framework offers placeholders that must be replaced by actual definitions in order to form a model. We will demonstrate how to do so in Sec. 4.3.

4.2.1 Strategy Profile, Final Graph, Cost

There are always these two parameters: the number of players $n \in \mathbb{N}$, usually $n \geq 3$, and the edge cost $\alpha > 0$. Each player v decides to which other vertices she would like to be connected by an edge. The requests of player v can be expressed as a vector $S_v \in \{0,1\}^n$. If $S_{vw} = 1$, then v would like the edge $\{v,w\}$ to exist. Such a vector S_v is called a *strategy* for player v. A vector of strategies $S = (S_v)_{v \in V}$, one for each player, is called a *strategy profile*. We can also denote a strategy profile as an $n \times n$ matrix $(S_{vw})_{v \in V, w \in V}$, then row number v is the strategy of player v. Strategy profiles correspond to the concept known as "action distribution" in Part I.

Let a strategy profile S be given. The graph which actually is built is called the *final graph*, denoted $G(S) = (V, E(S))$. The actual definition of the final graph is model-specific, i.e., one of the placeholders to be filled in to form a concrete model. We will consider the following two alternative definitions.

[1] Depending on the equilibrium concept, not all ways of deviating from the current decisions must be taken into consideration, or additional properties must be given for an equilibrium. Players might also not be allowed to make certain decisions unilaterally, e.g., v might need w's permission to build $\{v,w\}$. We will discuss different equilibrium concepts later.

4.1 Definition.

(i) We define the *unilateral* final graph[2] $G^{\mathsf{U}}(S) = (V, E^{\mathsf{U}}(S))$ by

$$E^{\mathsf{U}}(S) \coloneqq \{\{v,w\}\,;\; S_{vw} = 1 \vee S_{wv} = 1\}\ .$$

(ii) We define the *bilateral* final graph $G^{\mathsf{B}}(S) = (V, E^{\mathsf{B}}(S))$ by

$$E^{\mathsf{B}}(S) \coloneqq \{\{v,w\}\,;\; S_{vw} = 1 \wedge S_{wv} = 1\}\ .$$

Often we omit the "U" and "B" superscripts from our notation if a statement is supposed to hold for both cases or if it was explained earlier that we restrict to one of the two.

In the unilateral case, it suffices when *one* of the endpoints, v or w, requests an edge $\{v,w\}$ for it to appear in the final graph. In the bilateral case, *both* endpoints, v and w, have to request an edge $\{v,w\}$ for it to appear in the final graph. Denote $m^{\mathsf{U}}(S) \coloneqq |E^{\mathsf{U}}(S)|$ and $m^{\mathsf{B}}(S) \coloneqq |E^{\mathsf{B}}(S)|$ the number of edges in the final graphs.

Each player $v \in V$ experiences an *individual cost*, denoted $C_v(S,\alpha)$, comprised of *building cost* plus *indirect cost*. Building cost is different for the unilateral and bilateral cases:

– In the unilateral case, player v pays an amount of α for incident edges in the final graph that she has requested, i.e., v experiences building cost $\sum_{w\in V} S_{vw}\,\alpha$.

– In the bilateral case, for each edge $\{v,w\}$ in the final graph, both endpoints, v and w, are charged α, i.e., each v experiences building cost $\deg_{G(S)}(v)\,\alpha$.

For the bilateral case, it is also common in the literature to charge each player v an amount of α for every link that v requested, no matter whether the link was actually built, i.e., no matter whether it occurs in the final graph. So each v would be charged $\sum_{w\in V} S_{vw}\,\alpha$, as in the unilateral case. This makes players to pay even for useless requests. For the unilateral case, even with our notion of cost, it is possible that both endpoints pay for a link while it would have been sufficient if one of them paid for it; in other words, players may be charged for unnec-

[2] This definition is also known as the *closure* of S, see, e.g., [7, 8, 46, 82].

essary requests. We have chosen our notion of building cost since in both the unilateral and bilateral case, it guarantees at least that no-one is charged for a link that is not in the final graph. Ultimately, however, this does not matter since we will rule out unnecessary and useless requests soon when we restrict to so-called clean strategy profiles.

The indirect cost for player v is denoted $I_v(G(S))$ and only depends on the final graph. We require the indirect cost to not even depend on v itself, but only on v's position in the final graph. This is of importance in particular if $G(S)$ has symmetry. For instance, if $G(S)$ is a cycle, then all vertices experience the same indirect cost. If $G(S)$ is a path, then both endpoints experience the same indirect cost. The actual definition of indirect cost is model-specific; it is the second (and final) placeholder to be filled in to form a concrete model. For instance, we could choose the sum of distances to other vertices, i.e., $I_v(G(S)) := \sum_{w \in V} \mathrm{dist}_{G(S)}(v, w)$. Many other notions are possible. We will analyze a notion of indirect cost expressing robustness in Ch. 5.

The *social cost* (or sometimes just *cost*), denoted $C(S, \alpha)$, is the sum over all individual costs, i.e., $C(S, \alpha) := \sum_{v \in V} C_v(S, \alpha)$. The *total building cost* is the sum over the building costs of all players. The *total indirect cost* is the sum over the indirect costs of all players. Hence, social cost is total building cost plus total indirect cost.

Alternatively, instead of using indirect cost, one can also use a more positive notion. Instead of experiencing a cost $C_v(S, \alpha)$, player v can enjoy a *payoff* $\pi_v(S, \alpha)$. Payoff is a quantity called *income* (or *utility*, *benefit*) minus building cost. Income is the positive notion of indirect cost. Essentially, it is the same concept expressed differently. We stick to the notion of indirect cost.

In the unilateral case, strategy profiles can contain *unnecessary* requests, i.e., $S_{vw} = 1$ and $S_{wv} = 1$, while one of the two would have sufficed to have $\{v, w\}$ in the final graph. In the bilateral case, strategy profiles can contain *useless* requests, i.e., $S_{vw} = 1$ and $S_{wv} = 0$, while $\{v, w\}$ is not in the final graph even though v requested it. In many respects, it is easier to work without unnecessary or useless requests.

4.2 Definition.

(i) A strategy profile S is called *clean for the unilateral case* if for all $v, w \in V$ the following implication holds: $S_{vw} = 1 \implies S_{wv} = 0$.

(ii) A strategy profile S is called *clean for the bilateral case* if for all $v, w \in V$ the following implication holds: $S_{vw} = 1 \Longrightarrow S_{wv} = 1$.

We will restrict to clean strategy profiles later. For each strategy profile there exists a clean strategy profile with the same final graph and with the same or less individual cost for each player. The reason is simple: removing unnecessary or useless requests does not change the final graph, hence it does not change indirect cost. It cannot increase building cost for any player – in fact, it even reduces building cost for certain players in the unilateral case. Hence in the study of optima, we can restrict to clean strategy profiles. The same holds for all three kinds of equilibria introduced later, as we will explain there.

4.2.2 Graph-Related Notions

Most of the time, we use a graph-related language when speaking of strategy profiles and cost. We already know the notion of the final graph, but unfortunately it does not always capture all relevant information. This can be compensated for by using a kind of directed version of the final graph. For a strategy profile S, denote the directed graph $\vec{G}(S) = (V, \vec{E}(S))$ with

$$\vec{E}(S) \coloneqq \{(v, w);\ S_{vw} = 1\}\ .$$

Then S is completely determined by $\vec{G}(S)$ and vice versa – in fact, S is the adjacency matrix of $\vec{G}(S)$. We give an example. We refer to players by numbers $V = \{1, \ldots, 9\}$. Denote the strategy profile S by a matrix:

$$S \coloneqq \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{4.1}$$

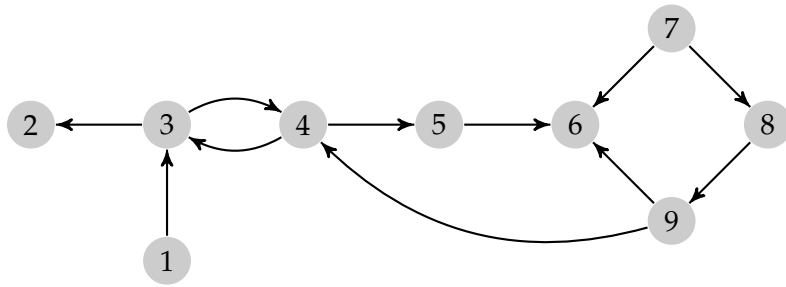


Figure 4.1. Strategy profile S from (4.1) represented as a directed graph $\vec{G}(S)$.

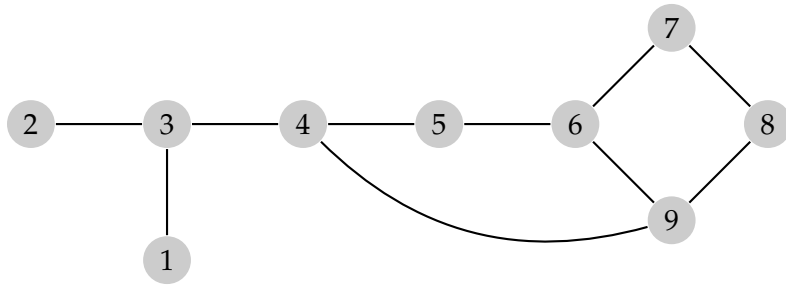


Figure 4.2. Final graph $G^{\mathsf{U}}(S)$ for strategy profile S from (4.1).

Then Fig. 4.1 shows $\vec{G}(S)$ and Fig. 4.2 shows $G^{\mathsf{U}}(S)$. Note that player 6 has three incident edges, although she did not issue any requests; row number 6 has only zeros in the matrix. Player 6 has zero building cost in the unilateral case. This strategy profile is not clean for the unilateral case, since 3 and 4 name each other. It is also not clean for the bilateral case, since, for instance, 1 names 3, but 3 does not name 1. So there is no link $\{1,3\}$ in $G^{\mathsf{B}}(S)$; in fact, the only link in $G^{\mathsf{B}}(S)$ is $\{3,4\}$.

Now we leave this example and consider the cost for a general strategy profile S, first in the unilateral case. The final graph $G^{\mathsf{U}}(S)$ is a modification of $\vec{G}(S)$ where the directions of links are forgotten, i.e., the underlying undirected graph of $\vec{G}(S)$. The building cost of v is v's out-degree in $\vec{G}(S)$ times α. We can write the individual cost of v as

$$C_v^{\mathsf{U}}(S,\alpha) = \deg^{\text{out}}_{\vec{G}(S)}(v)\,\alpha + I_v(G^{\mathsf{U}}(S)) \ .$$

If S is clean, we have the social cost:

$$C^{\mathsf{U}}(S,\alpha) = |E^{\mathsf{U}}(S)|\,\alpha + I(G^{\mathsf{U}}(S)) \ .$$

So everything can be seen relative to a directed graph $\vec{G} = \vec{G}(S)$; the social cost is even fully determined by the underlying undirected graph of $\vec{G}$ if S is clean.

Now consider the bilateral case. The situation is even simpler here: individual and social cost can be expressed fully in terms of the final graph. For a strategy profile S and player v the individual cost is:

$$C_v^{\mathsf{B}}(S,\alpha) = \deg_{G^{\mathsf{B}}(S)}(v)\,\alpha + I_v(G^{\mathsf{B}}(S)) \ ,$$

and the social cost:

$$C^{\mathsf{B}}(S,\alpha) = 2\,|E^{\mathsf{B}}(S)|\,\alpha + I(G^{\mathsf{B}}(S)) \ .$$

If S is clean, we can even determine S from knowing $G^{\mathsf{B}}(S)$. This often allows us to work with final graphs (which are undirected) instead of strategy profiles (or directed graphs) in the bilateral case. In the unilateral and bilateral case, the social cost of a clean strategy profile only depends on the final graph, and so we sometimes write $C(G,\alpha)$ instead of $C(S,\alpha)$, with $G = G(S)$.

We use a short notation for strategy profile changes. Let S be a strategy profile and $v, w \in V$. Then $S + (v,w)$ and $S - (v,w)$, respectively, are defined by

$$(S + (v,w))_{xy} \coloneqq \begin{cases} 1 & \text{if } (x,y) = (v,w) \\ S_{xy} & \text{otherwise} \end{cases} \ ,$$

$$(S - (v,w))_{xy} \coloneqq \begin{cases} 0 & \text{if } (x,y) = (v,w) \\ S_{xy} & \text{otherwise} \end{cases} \ ,$$

for all $x, y \in V$. In other words, $S + (v,w)$ means setting S_{vw} to 1, and $S - (v,w)$ means setting S_{vw} to 0. Inductively, we extend this definition to sets F of pairs $F \subseteq V \times V$. This is a similar notation as for directed graphs. Indeed, we have $\vec{G}(S + (v,w)) = \vec{G}(S) + (v,w)$

and $\vec{G}(S-(v,w)) = \vec{G}(S)-(v,w)$. This notation is useful for both the unilateral and the bilateral case. We have for $F = \{(v,w_1),\dots,(v,w_k)\}$ and a *clean* strategy profile S:

$$\begin{aligned}
G^{\mathsf{U}}(S+F) &= G^{\mathsf{U}}(S) + \{\{v,w_1\},\dots,\{v,w_k\}\} \\
G^{\mathsf{U}}(S-F) &= G^{\mathsf{U}}(S) - \{\{v,w_1\},\dots,\{v,w_k\}\} \\
G^{\mathsf{B}}(S+F) &= G^{\mathsf{B}}(S) \\
G^{\mathsf{B}}(S-F) &= G^{\mathsf{B}}(S) - \{\{v,w_1\},\dots,\{v,w_k\}\} \quad .
\end{aligned}$$

This shows that the two cases are similar regarding link deletion, but different regarding link creation. We use the notion of "v selling (or removing, deleting) an edge $\{v,w\}$" if v changes her strategy such that $\{v,w\}$ is no longer in the final graph. We use the notion of "v buying (or creating, adding) an edge $\{v,w\}$" if v changes her strategy such that $\{v,w\}$ is now in the final graph. The latter is impossible for the bilateral case and a clean strategy profile.

Convention. Strategy profiles in the following are assumed to be clean.

As mentioned earlier, this is without loss of generality for the study of optima and equilibria, hence in particular the prices of anarchy and stability.

4.2.3 Nash Equilibrium and Price of Anarchy

4.3 Definition.

– A strategy profile S is called a *Nash equilibrium* if

$$C_v(S+A-D,\alpha) \geq C_v(S,\alpha)$$

for all $v \in V$ and all $A, D \subseteq \{v\} \times V$.

– We call a directed graph $\vec{G} = (V,\vec{E})$ a Nash equilibrium if there exists a strategy profile S being a Nash equilibrium and $\vec{G} = \vec{G}(S)$.

– We call an undirected graph $G = (V,E)$ a Nash equilibrium if there exists a strategy profile S being a Nash equilibrium and $G = G(S)$.

In other words, S is a Nash equilibrium if no player can strictly improve her individual cost by changing her strategy, given the strategies of the other players. Denote the set of all Nash equilibria for given n and α by $\mathcal{N}(n,\alpha)$. The concept of Nash equilibrium is not well suited for the bilateral case; we will elaborate on this in Sec. 4.4.

With our definition of building cost, an unclean strategy profile cannot be a Nash equilibrium (in the unilateral case), since a player issuing unnecessary requests could strictly improve by removing those requests. Thus, our restriction to clean strategy profiles is justified.[3]

4.4 Definition.

- A strategy profile with minimum social cost for fixed n and α is called an *optimum*. The optimum social cost is denoted $\mathsf{OPT}(n,\alpha)$.
- We call a directed graph $\vec{G} = (V,\vec{E})$ an optimum if there exists a strategy profile S being an optimum and $\vec{G} = \vec{G}(S)$.
- We call an undirected graph $G = (V,E)$ an optimum if there exists a strategy profile S being an optimum and $G = G(S)$.

The optimum social cost can be different for the unilateral and bilateral case, and we write $\mathsf{OPT}^{\mathsf{U}}(n,\alpha)$ or $\mathsf{OPT}^{\mathsf{B}}(n,\alpha)$ to explicitly distinguish the two cases where necessary.

We always assume that the optimum social cost is positive. All our concrete models studied later will automatically fulfill this requirement.

4.5 Definition. We define the *price of stability* and the *price of anarchy* with respect to Nash equilibrium, respectively, by:

$$\sigma^{\mathsf{N}}(n,\alpha) := \min_{S\in\mathcal{N}(n,\alpha)} \frac{C(S,\alpha)}{\mathsf{OPT}(n,\alpha)}$$

and

$$\rho^{\mathsf{N}}(n,\alpha) := \max_{S\in\mathcal{N}(n,\alpha)} \frac{C(S,\alpha)}{\mathsf{OPT}(n,\alpha)} \quad .$$

We omit the "N" superscript if it was explained earlier that we restrict to Nash equilibrium.

[3] It would also be if we did not charge players for unnecessary requests, since then those do not influence cost at all.

4.3 A Concrete Model: Sum of Distances

To become familiar with the basic idea and the framework, we consider a concrete model, which we call the *sum-distance model*. It was the first such model for which a quantitative analysis of the price of anarchy was conducted, namely in 2003 by Fabrikant, Luthra, Maneva, Papadimitriou, and Shenker [41].

We stick to the unilateral case. The only placeholder left to fill in the framework is the indirect cost. We define indirect cost for player v as the sum of distances to all other players, i.e.,

$$I_v(G(S)) := \sum_{w \in V} \mathrm{dist}_{G(S)}(v, w) \ .$$

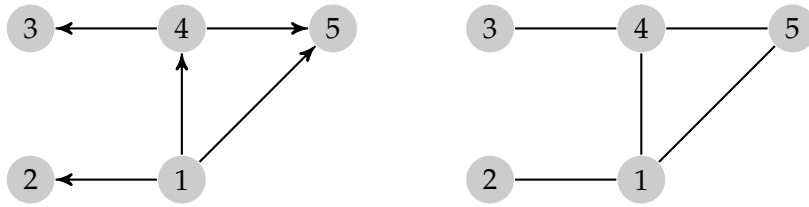
This indirect cost is suited to express routing cost: each player wishes to route one unit of traffic to each other player, and the cost for routing one unit along an edge is 1.

Recall that we can express strategy profiles as directed graphs and that we restrict to clean strategy profiles. We will make use of this extensively in this section and also use a streamlined notation. First consider the social cost. It only depends on the final graph, say, $G = (V, E)$. As common in graph theory, we do not carry the name of the concerning graph around in all our notation. For instance, we write "$\mathrm{dist}(v, w)$" instead of "$\mathrm{dist}_G(v, w)$". Denote $m = |E|$ the number of edges in the final graph. We then have a compact representation of the social cost:

$$C(G, \alpha) = m\,\alpha + \sum_{v,w \in V} \mathrm{dist}(v, w) \ .$$

Since we are in the unilateral case, it is sometimes important who pays for which edge. Then we use a directed version of G to express that, denoted $\vec{G} = (V, \vec{E})$, i.e., we choose $\vec{G}$ such that $\vec{G} = \vec{G}(S)$, where S is the strategy profile we wish to express. Then the number of edges for which v has to pay is her out-degree $\deg^{\mathrm{out}}(v)$, with respect to $\vec{G}$. We can then express the individual cost of player v in only graph-theoretic terms:

$$C_v(\vec{G}, \alpha) = \deg^{\mathrm{out}}(v)\,\alpha + \sum_{w \in V} \mathrm{dist}(v, w) \ .$$



a. Strategy profile written as a directed graph $\vec{G}$.

b. The corresponding final graph G: the link directions are simply forgotten.

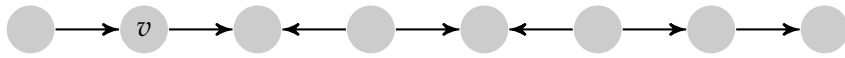
Figure 4.3. An example.

The distance "dist(v, w)" in the indirect cost is of course taken with respect to the undirected version, namely G.

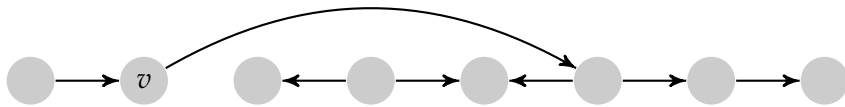
As an example, consider Fig. 4.3. We write out all individual costs for a general α:

$$\begin{aligned}
C_1(\vec{G}, \alpha) &= 3\alpha + 1 + 2 + 1 + 1 = 3\alpha + 5 \\
C_2(\vec{G}, \alpha) &= 0\alpha + 1 + 3 + 2 + 2 = 8 \\
C_3(\vec{G}, \alpha) &= 0\alpha + 2 + 3 + 1 + 2 = 8 \\
C_4(\vec{G}, \alpha) &= 2\alpha + 1 + 2 + 1 + 1 = 2\alpha + 5 \\
C_5(\vec{G}, \alpha) &= 0\alpha + 1 + 2 + 2 + 1 = 6
\end{aligned}$$

Since $\text{dist}(v, w) = \infty$ if v and w are in different connected components, optima and Nash equilibria are connected. Nash equilibrium has a rather strong requirement. It does not only mean that no player has an incentive to add one or more edges or to remove one or more edges. It also means that no player has an incentive to do an *exchange* of edges, i.e., to remove one or more edges *and* to add one or more other edges instead. The difference can be seen by an example. Consider a path on n vertices with arbitrary edge directions and, for simplicity, $\alpha \geq \frac{(n-1)\,n}{2}$. Then α is at least the largest individual indirect cost, so there is no incentive for any player to buy one or more edges. There is also no incentive to sell any edges, since that would make the graph disconnected. Hence, if exchanges were not allowed, we would have a Nash equilibrium. However, since exchanges *are* allowed, we do not have a Nash equilibrium: most of the vertices can sell their one edge



a. A path on n vertices.



b. Player v exchanges her edge.

Figure 4.4. If no exchanges were allowed, the path would be a Nash equilibrium if α is large enough. In Fig. 4.4b, we see how player v can improve her cost from $\alpha + 22$ to $\alpha + 16$ by exchanging an edge.

(or one of their edges) and reconnect to the truncated part of the path in a more efficient way, as shown in Fig. 4.4.

We will now analyze the sum-distance model to some extent. We will not present all state-of-the-art results and techniques here, but merely consider the $\max\{1, O(\sqrt{\alpha})\}$ bound on the price of anarchy, which was first proved in [41]. We partly follow [41] and [85] in our presentation. An overview of further results is given in Sec. 4.5. Let $n \geq 3$ in the following.

4.6 Proposition.

(i) If $\alpha \leq 2$, then the complete graph K_n is optimal.
It has social cost $\frac{n(n-1)}{2}\alpha + n(n-1) = \Theta(n^2\alpha + n^2)$.

(ii) If $\alpha \geq 2$, then a star is optimal.
It has social cost $(n-1)\alpha + 2(n-1)^2 = \Theta(n\alpha + n^2)$.

Proof. Recall that the social cost is fully determined by the final graph, so we can restrict to undirected graphs for considering optima. We have $\mathrm{dist}(v, w) \geq 2$ for each pair $v, w \in V$, $v \neq w$ that is *not* connected by an edge. Therefore we have for any connected $G = (V, E)$ with $m = |E|$ the social cost lower-bounded:

$$\begin{aligned} C(G, \alpha) &= m\alpha + \sum_{v,w \in V} \mathrm{dist}(v, w) = m\alpha + 2m + \sum_{\substack{v,w \in V \\ \{v,w\} \notin E}} \mathrm{dist}(v, w) \\ &\geq m\alpha + 2m + 2\left(n(n-1) - 2m\right) = m(\alpha - 2) + 2n(n-1) \ . \end{aligned} \tag{4.2}$$

This inequality is tight if $\operatorname{diam}(G) \leq 2$.

(i) If $\alpha \leq 2$, then $\alpha - 2 \leq 0$ and so the right-hand-side of (4.2) is non-increasing in m, so it is minimal for maximum m. For the graph with maximum m, the K_n, the inequality is tight, so K_n is optimal.

(ii) If $\alpha \geq 2$, then $\alpha - 2 \geq 0$ and so the right-hand-side of (4.2) is non-decreasing in m, so it is minimal for minimum m. All trees have minimum m, namely $m = n - 1$. So, a tree for which the inequality is tight is optimal. A star has diameter 2, and so the inequality is tight and hence a star is optimal. Its social cost also follows from (4.2). □

4.7 Proposition.

(i) If $\alpha \leq 1$, then any S for which $G(S) = K_n$, is a Nash equilibrium. Moreover, if $\alpha < 1$, then these are the only Nash equilibria.

(ii) If $\alpha \geq 1$, then any S for which $G(S)$ is a star, is a Nash equilibrium.

Proof. (i) Let $\alpha \leq 1$. If a player v sells an edge, her indirect cost rises by at least 1. So if $\alpha \leq 1$, there is no incentive to sell any edges. If the final graph is K_n, then selling is the only option players have. Hence any strategy profile that has K_n as final graph is a Nash equilibrium.

If we have a graph with diameter at least 2, then there is a player who can reduce her indirect cost by at least 1. So, if $\alpha < 1$, no Nash equilibrium can have diameter more than 1.

(ii) Let $\alpha \geq 1$. Let $\vec{G} = (V, \vec{E})$ be a strategy profile such that the final graph G is a star. The center vertex cannot buy additional edges. If an outer vertex buys one or more additional edges, then her indirect cost decreases by exactly 1 per new edge, since it reduces the distance to exactly one vertex from 2 to 1. Since $\alpha \geq 1$, there is hence no incentive to buy any more edges. If we assume that all edges point outward in $\vec{G}$, we are done here. This is because the center vertex is the only one that could sell these edges, but then the graph would become disconnected. The center vertex has no way of selling an edge and making the graph connected again by adding a different edge.

However, we can even show that it does not matter who owns the edges in the star. Let c be the center vertex and v be an outer vertex that owns edge $e = \{c, v\}$, i.e., $(v, c) \in \vec{E}$. Selling e and not buying any new edges is impossible for v, since that would make the graph

disconnected. This leaves us to consider whether it is beneficial for v to sell e and to buy edges of the form $\{v, w_0\}, \ldots, \{v, w_k\}$, with $w_0, \ldots, w_k$ being outer vertices distinct from v.

Denote $e_i = \{v, w_i\}$, for $i = 0, \ldots, k$. Consider e being sold and then the edges $e_0, \ldots, e_k$ being bought one-by-one. When e is sold and e_0 bought, the building cost does not change. The indirect cost changes: v is now 1 edge closer to w_0, but also 1 edge farther from all other $n-2$ vertices. So we note an increase of $n-3 \geq 0$ in indirect cost, so the strategy change was not beneficial so far. Now e_1 is bought. This reduces distance only to w_1, namely by 1. So the indirect cost reduces by 1, but the building cost increases by $\alpha \geq 1$. So the strategy change is still not beneficial. The same holds for all further steps, where $e_2, \ldots, e_k$ are bought. We conclude that there is no beneficial strategy change possible for v, and hence we have a Nash equilibrium. □

Now that we know some optima and Nash equilibria, we can easily bound the price of stability.

4.8 Theorem. *The price of stability is no more than $\frac{4}{3}$, for any n and α.*

Proof. By Prop. 4.6 and Prop. 4.7, the price of stability is 1 for $\alpha \leq 1$ and $\alpha \geq 2$: if $\alpha \leq 1$, then the complete graph is optimal and also a Nash equilibrium, and if $\alpha \geq 2$ then the same is the case for a star. In the range $1 < \alpha < 2$, we get a bound on the price of stability by dividing the social cost of a star by that of the complete graph:

$$\begin{aligned}\frac{(n-1)\,\alpha + 2\,(n-1)^2}{\frac{n\,(n-1)}{2}\alpha + n\,(n-1)} &= \frac{1}{n}\,\frac{\alpha + 2\,(n-1)}{\frac{1}{2}\alpha + 1} \\ &\leq \frac{1}{n}\,\frac{2 + 2\,(n-1)}{\frac{1}{2} + 1} = \frac{1}{n}\,\frac{2n}{\frac{3}{2}} = \frac{4}{3}\ .\end{aligned}$$ □

Now we aim for a bound on the price of anarchy. From previous propositions, we already know that it is 1 for $\alpha < 1$. To have a bound for larger α, we look at the diameter of Nash equilibria.

4.9 Lemma. *The diameter of a Nash equilibrium is at most* $\max\{3, 2\sqrt{\alpha}\}$.

Proof. Let $P = (v_0, \ldots, v_\ell)$ be a shortest path between some vertices $v := v_0$ and $w := v_\ell$. If $\ell \leq 3$, we are done. So assume $\ell \geq 4$. We show that $\ell \leq 2\sqrt{\alpha}$. We first treat the case that ℓ is even. By building $\{v, w\}$, player v reduces the distance to v_i to $\min\{i, \ell - i + 1\}$ for each $i \in [\ell]$. The original distance to v_i is i, so the improvement is positive if $i > \ell - i + 1$, i.e., $i > \frac{\ell}{2} + \frac{1}{2}$, that is if $i \in \left\{\frac{\ell}{2} + 1, \ldots, \ell\right\}$ since ℓ is even. Total savings in indirect cost are at least:

$$\sum_{i=1}^{\ell} \left(i - \min\{i, \ell - i + 1\}\right) = \sum_{i=\frac{\ell}{2}+1}^{\ell} \left(i - (\ell - i + 1)\right)$$

$$= \sum_{i=\frac{\ell}{2}+1}^{\ell} (2i - \ell - 1) = \sum_{i=1}^{\ell-\frac{\ell}{2}} \left(2\left(i + \frac{\ell}{2}\right) - \ell - 1\right)$$

$$= \sum_{i=1}^{\frac{\ell}{2}} (2i - 1) = 2\,\frac{\frac{\ell}{2}\left(\frac{\ell}{2} + 1\right)}{2} - \frac{\ell}{2} = \left(\frac{\ell}{2}\right)^2 .$$

The actual savings may be higher, since the distance to other vertices than those on P may be reduced as well. But for our result the above estimation is good enough. Since we have a Nash equilibrium, the improvement in indirect cost can be at most α, and so $\ell \leq 2\sqrt{\alpha}$.

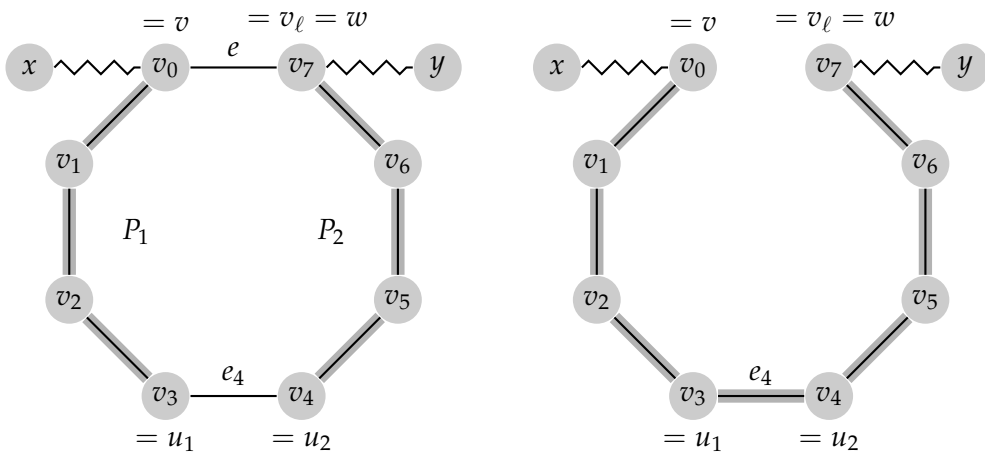
If ℓ is odd, i.e., $\ell \geq 5$, then we consider the savings resulting for v from building $\{v, v_{\ell-1}\}$. Since $\ell - 1$ is even, we already know that we have savings of at least $(\frac{\ell-1}{2})^2$. This is augmented by savings equal to $\ell - 2$, since v now can reach w via a path of length 2 instead of ℓ. In total, savings are at least

$$\left(\frac{\ell - 1}{2}\right)^2 + \ell - 2 = \frac{1}{4}\left(\ell^2 - 2\ell + 1\right) + \ell - 2$$

$$= \left(\frac{\ell}{2}\right)^2 - \frac{\ell}{2} + \ell - \frac{7}{4} = \left(\frac{\ell}{2}\right)^2 + \frac{\ell}{2} - \frac{7}{4} \geq \left(\frac{\ell}{2}\right)^2 .$$

We need $\ell \geq 5$ in the last step. It follows $\ell \leq 2\sqrt{\alpha}$. □

The lemma gives us a bound on the indirect cost part of the social cost of any Nash equilibrium, namely $O(n^2\sqrt{\alpha})$, for $\alpha \geq 1$. Next we relate the number of edges to the diameter. This is accomplished by



a. A shortest path from x to y via e is shown at the top with $(x, \dots, v)$ and $(w, \dots, y)$ each depicted as a zig-zag path. Below is a shortest cycle containing e, here of length $\ell + 1 = 8$. Paths P_1 and P_2 are highlighted. We have $|P_1|, |P_2| \leq \operatorname{diam}(G)$.

b. How we patch the shortest path when e was removed. The replacement part is highlighted. The new path is at most $2\operatorname{diam}(G)$ longer.

Figure 4.5. Distances increase by no more than $2\operatorname{diam}(G)$.

the following proposition and lemma.

4.10 Proposition. *Let $G = (V, E)$ be any graph and $e = \{v, w\} \in E$ a non-bridge. Then the removal of e does not increase the distance between any two vertices by more than $2\operatorname{diam}(G)$.*

Proof. Since $e = \{v, w\}$ is not a bridge, there is at least one cycle containing e. Let $C = (v_0, e_1, v_1, \dots, v_\ell, e_{\ell+1}, v_{\ell+1})$ be a cycle with $v = v_0 = v_{\ell+1}$, $w = v_\ell$, $e = e_{\ell+1}$, chosen so that it has minimum length $\ell + 1$ among all cycles containing e. Let $P_1 \coloneqq (v_0, \dots, u_1)$ with $u_1 \coloneqq v_{\lfloor \frac{\ell}{2} \rfloor}$ and $P_2 \coloneqq (v_\ell, e_\ell, v_{\ell-1}, \dots, u_2)$ with $u_2 \coloneqq v_{\lceil \frac{\ell}{2} \rceil}$. If ℓ is even, then the concatenation $R \coloneqq P_1 \circ P_2^{-1}$ connects v with w without using e. If ℓ is odd, then $R \coloneqq P_1 \circ (u_1, e_{\lceil \frac{\ell}{2} \rceil}, u_2) \circ P_2^{-1}$ does the same. Fig. 4.5 shows the situation for odd ℓ. In any case we have a 'replacement' R for e with $|R| \leq |P_1| + |P_2| + 1$, so the distance between any two vertices grows at most by $|P_1| + |P_2|$.

We are left to bound this against the diameter. We have $|P_1| = |P_2| = \lfloor \frac{\ell}{2} \rfloor$. Let P be a shortest path from v to u_1 that is shorter than P_1. Since C is chosen as short as possible, P must start with e. The rest of P cannot be shorter than P_2, also because of the choice of C. Thus $|P_1| > |P| \geq 1 + |P_2|$, so $\lfloor \frac{\ell}{2} \rfloor > 1 + \lfloor \frac{\ell}{2} \rfloor$, a contradiction. The same argument holds with P_1 and P_2 exchanged. It follows that P_1 as well as P_2 are shortest paths, so $|P_1|, |P_2| \leq \operatorname{diam}(G)$. □

4.11 Lemma. *Let $\vec{G} = (V, \vec{E})$ be a Nash equilibrium and $d \coloneqq \operatorname{diam}(G)$. Then the number of edges in G is bounded by $O(n + \frac{n^2 d}{\alpha})$.*

Proof. There are at most $n - 1$ bridges, contributing the first term in the bound. Let us consider non-bridges, i.e., edges that can be removed without destroying connectivity. Fix $v \in V$ and let $\{v, w_1\}, \ldots, \{v, w_k\}$ be all non-bridges for which v pays, i.e., $(v, w_1), \ldots, (v, w_k) \in \vec{E}$. For each $i \in [k]$ let V_i be the set of vertices w such that all shortest v-w paths lead via $e_i \coloneqq \{v, w_i\}$. By Prop. 4.10, deleting e_i increases the distance to no vertex by more than $2d$. Since v still chose to build e_i, there must be 'many' vertices to which e_i reduces the distance to v, so that the investment of α pays off. The vertices for which e_i reduces the distance to v are exactly those in V_i. Precisely, we have $\alpha \leq |V_i|\, 2d$. This holds for each i. It follows a bound on the building cost of v:

$$k\alpha = \sum_{i=1}^{k} \alpha \leq \sum_{i=1}^{k} |V_i|\, 2d \leq 2nd \ .$$

Summing this over all n players yields a bound of $O(n^2 d)$ on the total building cost for non-bridges, and hence the claimed bound on the number of edges. □

4.12 Lemma. *The price of anarchy is bounded by the largest diameter of a final graph, that is $\rho(n, \alpha) = O(\max_{\vec{G} \in \mathcal{N}(n,\alpha)} \operatorname{diam}(G))$, for all n and α.*

Proof. For $\alpha < 1$, it follows directly from Prop. 4.6(i) and Prop. 4.7(i) that the price of anarchy is 1.

Let $\alpha \geq 1$. Then Prop. 4.6 gives a lower bound of $\Theta(n\alpha + n^2)$ on the optimal social cost. Denote $d \coloneqq \max_{\vec{G} \in \mathcal{N}(n,\alpha)} \operatorname{diam}(G)$. By Lem. 4.11, we have an $O((n + \frac{n^2 d}{\alpha})\, \alpha) = O(n\alpha + n^2 d)$ bound on the total building

cost of any Nash equilibrium from $\mathcal{N}(n,\alpha)$. We always have an $O(n^2 d)$ bound on the total indirect cost. It follows

$$\rho(n,\alpha) = O\left(\frac{n\alpha + n^2 d}{n\alpha + n^2}\right) = O(d) \ . \qquad \square$$

4.13 Theorem.

(i) If $\alpha < 1$, the price of anarchy is 1.

(ii) If $\alpha \geq 1$, the price of anarchy is $O(\sqrt{\alpha})$.

(iii) If $\alpha > \frac{(n-1)\,n}{2} = \Omega(n^2)$, the price of anarchy is $O(1)$.

Proof. (i) Follows directly from Prop. 4.6(i) and Prop. 4.7(i). (As in the previous proof.)

(ii) Follows directly from Lem. 4.9 and Lem. 4.12.

(iii) The worst increase in indirect cost by selling a non-bridge is clearly bounded by the maximum conceivable individual indirect cost. The latter is attained if the graph is a path and for a player at one of its ends, namely $\sum_{k=1}^{n-1} k = \frac{(n-1)\,n}{2}$. So, if α is larger than this, there can be no non-bridges in a Nash equilibrium, and hence all Nash equilibria are trees. A tree has its social cost roughly bounded by $O(n\alpha + n^3)$. Dividing this by the social cost of a star, namely $\Theta(n\alpha + n^2)$, yields the $O(1)$ bound since $n\alpha = \Omega(n^3)$. $\square$

For the next result we only give the proof idea. It was proved independently by Lin [57] and Albers et al. [2]; a proof is also given in [85].

4.14 Theorem. *The price of anarchy is $O(1 + \frac{\alpha}{\sqrt{n}})$. In particular, it is $O(1)$ for $\alpha = O(\sqrt{n})$.*

Proof Idea. Carefully bound the diameter of an arbitrary Nash equilibrium, achieving a better bound than in Lem. 4.9, and then invoke Lem. 4.12. $\square$

4.4 More on the Bilateral Case

We have worked in the unilateral case in the previous section: the wish of one endpoint, v or w, is enough in order to have $\{v,w\}$ built. In the

bilateral case, *both* endpoints, v and w, have to request an edge $\{v, w\}$ for it to appear in the final graph. We will elaborate more on the bilateral case in this section. In Sec. 4.4.1 we consider equilibrium concepts that are better suited for the bilateral case than Nash equilibrium is. In Sec. 4.4.2 we revisit the sum-distance model in the bilateral case and discuss differences to the unilateral one. In Sec. 4.4.3 we show how simple-structured equilibria can be transfered from the unilateral to the bilateral case in general.

By definition of individual cost, each of the endpoints has to pay α for the edge in the bilateral case. This models the situation that establishing a link incurs a *non-transferable* cost at each end, and so requires the consent of both ends. If these costs were transferable, it would probably be reasonable to allow one endpoint to pay for both and have the edge built unilaterally, if desired. So, each player v would have the option to pay 2α and have an edge $\{v, w\}$ in the final graph even though, for whatever reason, player w does not request this edge. Hence the bilateral case is justified only when building cost is not transferable.

4.4.1 Bilateral Equilibrium Concepts

It is important to realize that in the bilateral case, a player can remove links unilaterally (i.e., by only changing her strategy) but there is no way for a player to unilaterally build a link (since we restrict to clean strategy profiles). The bilateral case requires us to rethink our equilibrium concept. Recall that Nash equilibrium means that no player can improve her cost by changing her strategy, *given the strategies of all other players*. It follows that the empty strategy profile, i.e., $S_{vw} = 0$ for all $v, w \in V$, is a Nash equilibrium. This may not be obvious, but it is true: by changing her strategy, each v cannot lower her indirect cost, since she cannot influence the final graph when all other players chose 0-vectors as their strategies. The empty strategy profile being an equilibrium independently of any other consideration appears unreasonable; for the sum-distance model it would trivially push the price of anarchy to ∞. Therefore, we add one more condition to the definition of Nash equilibrium. This condition introduces a minimum

of cooperation: if for each of the two endpoints the new link would be no impairment – or even beneficial – then it shall be built.

4.15 Definition.

– A strategy profile S is called a *pairwise Nash equilibrium* if it is a Nash equilibrium and for all $v, w \in V$ such that $\{v, w\} \notin E(S)$ the following implication holds:

$$\begin{aligned} &C_v(S + (v,w) + (w,v), \alpha) \leq C_v(S, \alpha) \\ &\implies C_w(S + (v,w) + (w,v), \alpha) > C_w(S, \alpha) \ . \end{aligned} \tag{4.3}$$

– We call a directed graph $\vec{G} = (V, \vec{E})$ a pairwise Nash equilibrium if there exists a strategy profile S being a pairwise Nash equilibrium and $\vec{G} = \vec{G}(S)$.

– We call an undirected graph $G = (V, E)$ a pairwise Nash equilibrium if there exists a strategy profile S being a pairwise Nash equilibrium and $G = G(S)$.

In other words, absence of $\{v, w\}$ in the final graph requires additional justification: if adding the edge, i.e., changing from S to $S + (v,w) + (w,v)$, is no impairment for one end, then it is an impairment for the other end. With this extension, the empty strategy profile is ruled out as equilibrium in any model that assigns indirect cost ∞ to a disconnected final graph. In fact, a pairwise Nash equilibrium is always connected then. This is so since additional links cannot be an impairment for anyone as long as cost is ∞: it remains ∞ or drops to some finite value, neither of which is an impairment.

It is also common in the literature to define pairwise Nash equilibrium with strict inequality on the left-hand side, i.e.,

$$\begin{aligned} &C_v(S + (v,w) + (w,v), \alpha) < C_v(S, \alpha) \\ &\implies C_w(S + (v,w) + (w,v), \alpha) > C_w(S, \alpha) \ . \end{aligned}$$

With this definition, the above argument does not work anymore. It does work, however, e.g., in the sum-distance model with a small modification: we set $\mathrm{dist}(v, w)$ not to ∞ but to a large number, strictly greater than $1 + \alpha$.[4] This modified model is essentially equivalent to

[4] A variant of the sum-distance model assigning a finite distance β to pairs of dis-

the original one, for the unilateral case with Nash equilibrium. Then adding a link connecting two components, or adding it to the empty graph for that matter, is beneficial for both endpoints, and so disconnected graphs are no pairwise Nash equilibria. We will stick to the definition using non-strict inequality.

Regarding the restriction to clean strategy profiles, the same reasoning given for Nash equilibria in the unilateral case on p. 102 applies. Additionally, we have to realize that removal of useless requests cannot create additional possibilities for forming links unilaterally – on the contrary, it eliminates such possibilities. Removing useless requests can hence not threaten the Nash equilibrium property of a pairwise Nash equilibrium.

Astonishingly, pairwise Nash equilibrium is equivalent – under additional requirements on the individual cost functions – to a seemingly much simpler concept, given in the next definition.

4.16 Definition.

– A strategy profile S is called *pairwise stable* if for all $v, w \in V$ the following two conditions hold:

(i) If $\{v, w\} \in E(S)$, then removal of the edge does not improve cost for any of the two endpoints v or w, i.e.,

$$\begin{aligned} C_v(S - (v,w), \alpha) &\geq C_v(S, \alpha) \\ \text{and} \quad C_w(S - (w,v), \alpha) &\geq C_w(S, \alpha) \ . \end{aligned}$$

(ii) If $\{v, w\} \notin E(S)$, then if adding the edge is no impairment to v, then it *is* an impairment to w, i.e.,

$$\begin{aligned} \text{if} \quad & C_v(S + (v,w) + (w,v), \alpha) \leq C_v(S, \alpha) \\ \text{then} \quad & C_w(S + (v,w) + (w,v), \alpha) > C_w(S, \alpha) \ . \end{aligned}$$

– We call a directed graph $\vec{G} = (V, \vec{E})$ pairwise stable[5] if there exists a strategy profile S being pairwise stable and $\vec{G} = \vec{G}(S)$.

connected players has been studied in [20], including the case $\beta \leq 1 + \alpha$. The concept of having a finite value for disconnection is also used in [56], called "disconnection penalty" there.

[5] This will hardly occur in practice, since pairwise stability can already be recognized by looking at the final graph.

– We call an undirected graph $G = (V, E)$ pairwise stable if there exists a strategy profile S being pairwise stable and $G = G(S)$.

Condition (ii) is exactly the additional condition that distinguishes the concept of Nash equilibrium from that of pairwise Nash equilibrium as per Def. 4.15. Pairwise stability is comfortable to work with, since one only has to consider single-edge changes. Moreover, it suffices to look at the final graph $G(S)$ to decide whether S is pairwise stable. In other words, pairwise stability can be seen as a property of graphs, without requiring the notions of strategies, strategy profiles, or players.

The restriction to clean strategy profiles is again justified, since with our definition of building cost the pairwise stability condition is not influenced by useless requests.[6]

4.17 Proposition. *A pairwise Nash equilibrium is pairwise stable.*

Proof. Def. 4.16(i) is implied by the Nash condition. Def. 4.16(ii) is directly implied, since it is part of the definition of pairwise Nash equilibrium. □

Denote $\mathcal{S}(n)$ all strategy profiles for n players. We introduce notation for the price of anarchy relative to different equilibrium concepts:

$$\rho^{\mathsf{PN}}(n,\alpha) := \max_{\substack{S\in\mathcal{S}(n)\\ S \text{ is a pairwise Nash eq.}}} \frac{C(S,\alpha)}{\mathsf{OPT}(n,\alpha)}$$

$$\rho^{\mathsf{PS}}(n,\alpha) := \max_{\substack{S\in\mathcal{S}(n)\\ S \text{ is pairwise stable}}} \frac{C(S,\alpha)}{\mathsf{OPT}(n,\alpha)}$$

We omit the "PN" and "PS" superscripts if it was explained earlier that we restrict to the respective equilibrium concept.

4.18 Proposition. *For any n and α we have $\rho^{\mathsf{PN}}(n,\alpha) \leq \rho^{\mathsf{PS}}(n,\alpha)$.*

Proof. Follows from Prop. 4.17. □

[6] This still holds if we charge players for useless requests. However, if we do so, there might in fact emerge more costly pairwise stable strategy profiles if we dropped the restriction to clean ones. Still it would be reasonable to maintain this restriction.

Fortunately, pairwise stability is even *equivalent* to pairwise Nash if cost functions are convex. For the sake of a streamlined notation, we fix α now and write C_v instead of $C_v(\cdot, \alpha)$ for the rest of this section.

4.19 Definition.

– Let $v \in V$ and S a strategy profile. The cost function C_v is called *convex in* S if for all $\{w_1, \ldots, w_k\} \subseteq V$ we have

$$\begin{aligned} &C_v\big(S - (v, w_1) - \ldots - (v, w_k)\big) - C_v(S) \\ \geq &\sum_{i=1}^{k} \big(C_v(S - (v, w_i)) - C_v(S)\big) \ . \end{aligned}$$

– The cost function C_v is called *convex* on a set of strategy profiles $\mathcal{S}$, if it is convex in every $S \in \mathcal{S}$.

– The cost function C_v is called *convex* if it is convex on $\mathcal{S}(n)$, i.e., the set of all strategy profiles for the given number n of players.

In other words, convexity means that removal of all the edges $\{v, w_1\}, \ldots, \{v, w_k\}$ increases cost for v by at least the sum of cost increments for each removal alone. When proving or disproving convexity, we may restrict all considerations to indirect cost since the convexity condition is equivalent to

$$\begin{aligned} &I_v\big(G(S - (v, w_1) - \ldots - (v, w_k))\big) - I_v(G(S)) \\ \geq &\sum_{i=1}^{k} \big(I_v(G(S - (v, w_i))) - I_v(G(S))\big) \ . \end{aligned}$$

4.20 Lemma. *Let S be a strategy profile. If C_v is convex in S for each $v \in V$, then S is a pairwise Nash equilibrium if and only if S is pairwise stable.*

Proof. One direction is Prop. 4.17.

Let now S be pairwise stable. We have to show that S is a Nash equilibrium. Fix a player $v \in V$. There is no way that v could unilaterally add an edge. So we only have to show that v has no incentive to delete edges. Let $\{w_1, \ldots, w_k\} \subseteq V$. By pairwise stability, there is no incentive to remove a single edge $\{v, w_i\}$, $i \in [k]$, i.e., $C_v(S - (v, w_i)) - C_v(S) \geq 0$ for each $i \in [k]$. By convexity, it follows

$$C_v(S - (v, w_1) - \ldots - (v, w_k)) - C_v(S)$$

$$\geq \sum_{i=1}^{k} \big(C_v(S-(v,w_i)) - C_v(S)\big) \geq 0 \ ,$$

and so v also has no incentive to remove all edges $\{v,w_1\},\ldots,\{v,w_k\}$. □

4.21 Corollary. *Let C_v for each v be convex (at least) on the set of pairwise stable strategy profiles. Then pairwise Nash equilibrium and pairwise stability are equivalent.*

The following lemma is due to Corbo and Parkes [32], their proof is based on a result by Calvó-Armengol and İlkiliç [23].

4.22 Lemma. *The individual cost functions in the sum-distance model from Sec. 4.3 are convex.*

Proof. Fix $v \in V$ and let $w_1,\ldots,w_k \in V$. We do induction on k. The case $k=1$ is trivial, so let $k>1$ and with $S' := S-(v,w_1)-\ldots-(v,w_{k-1})$ assume

$$C_v(S') - C_v(S) \geq \sum_{i=1}^{k-1} \big(C_v(S-(v,w_i)) - C_v(S)\big) \ .$$

For any strategy profile T and edge $\{v,w\} \in E(T)$ denote

$$V_{\{v,w\}}(T) := \{u \in V;\ \text{all shortest paths from } v \text{ to } u \text{ in } G(T) \text{ go via } \{v,w\}\} \ .$$

Switching from T to $T-(v,w)$ increases distance to all vertices in $V_{\{v,w\}}$. For all $\{v,w\} \in E(S')$ we have $V_{\{v,w\}}(S') \supseteq V_{\{v,w\}}(S)$. So switching from S' to $S'-(v,w_k)$ affects at least all those vertices that are affected by a switch from S to $S-(v,w_k)$. Let $u \in V_{\{v,w_k\}}(S)$. After removal of $\{v,w_k\}$, a shortest path from v to u must go via a different edge incident to v. There are fewer alternatives for this in S' than in S. Hence the effect for each such vertex u of switching from S' to $S'-(v,w_k)$ is at least as strong as when switching from S to $S-(v,w_k)$. It follows

$$C_v(S'-(v,w_k)) - C_v(S') \geq C_v(S-(v,w_k)) - C_v(S) \ . \tag{4.4}$$

By induction, we receive:

$$\begin{aligned}
&C_v(S-(v,w_1)-\ldots-(v,w_k))-C_v(S)\\
&=C_v(S'-(v,w_k))-C_v(S)\\
&=C_v(S'-(v,w_k))-C_v(S)+C_v(S')-C_v(S')\\
&=C_v(S'-(v,w_k))-C_v(S')+C_v(S')-C_v(S)\\
&\geq C_v(S-(v,w_k))-C_v(S)+C_v(S')-C_v(S) && \text{by (4.4)}\\
&\geq C_v(S-(v,w_k))-C_v(S)\\
&\quad+\sum_{i=1}^{k-1}\left(C_v(S-(v,w_i))-C_v(S)\right) && \text{by induction}\\
&=\sum_{i=1}^{k}\left(C_v(S-(v,w_i))-C_v(S)\right)\ . && \square
\end{aligned}$$

4.23 Theorem. *For the bilateral sum-distance model,* $\rho^{\mathsf{PN}}(n,\alpha)=\rho^{\mathsf{PS}}(n,\alpha)$ *for all* n *and* α.

Proof. Follows from Cor. 4.21 and Lem. 4.22. □

4.4.2 Lower Bound for Bilateral Sum-Distance Model

We study the *bilateral* sum-distance model using pairwise stability as equilibrium concept, which is equivalent to pairwise Nash equilibrium as shown in the previous section. Recall from Thm. 4.14 that the price of anarchy is $O(1)$ in the *unilateral* sum-distance model if $\alpha=O(\sqrt{n})$. We show that this bound no longer holds in the bilateral case, following Demaine et al. [38]. We first consider optima.

4.24 Proposition.

(i) If $\alpha\leq 1$, *then the complete graph* K_n *is optimal.*

(ii) If $\alpha\geq 1$, *then a star is optimal.*
It has social cost $2(n-1)\alpha+2(n-1)^2=\Theta(n\alpha+n^2)$.

Proof. We proceed as in the proof of Prop. 4.6, with the only difference that each edge has to be paid for the amount of α by both endpoints.

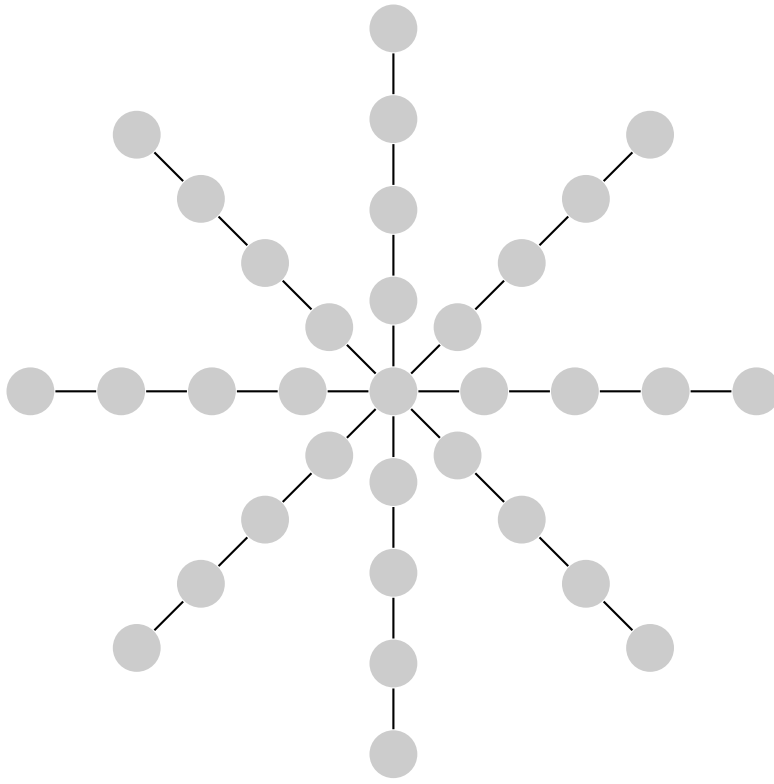


Figure 4.6. Generalized star with $\ell = 4$ and $N = 8$.

We have the social cost lower-bounded:

$$C(G, \alpha) \geq m\,(2\alpha - 2) + 2n\,(n-1) = 2m\,(\alpha - 1) + 2n\,(n-1)\ . \quad (4.5)$$

This inequality is tight if $\mathrm{diam}(G) \leq 2$.

(i) If $\alpha \leq 1$, then $\alpha - 1 \leq 0$ and so the right-hand-side of (4.5) is non-increasing in m, so it is minimal for maximum m. For the graph with maximum m, the K_n, the inequality is tight, so K_n is optimal.

(ii) If $\alpha \geq 1$, then $\alpha - 1 \geq 0$ and so the right-hand-side of (4.5) is non-decreasing in m, so it is minimal for minimum m. All trees have minimum m, namely $m = n - 1$. So, a tree for which the inequality is tight is optimal. A star has diameter 2, and so the inequality is tight and a star is optimal. Its social cost follows from (4.5). □

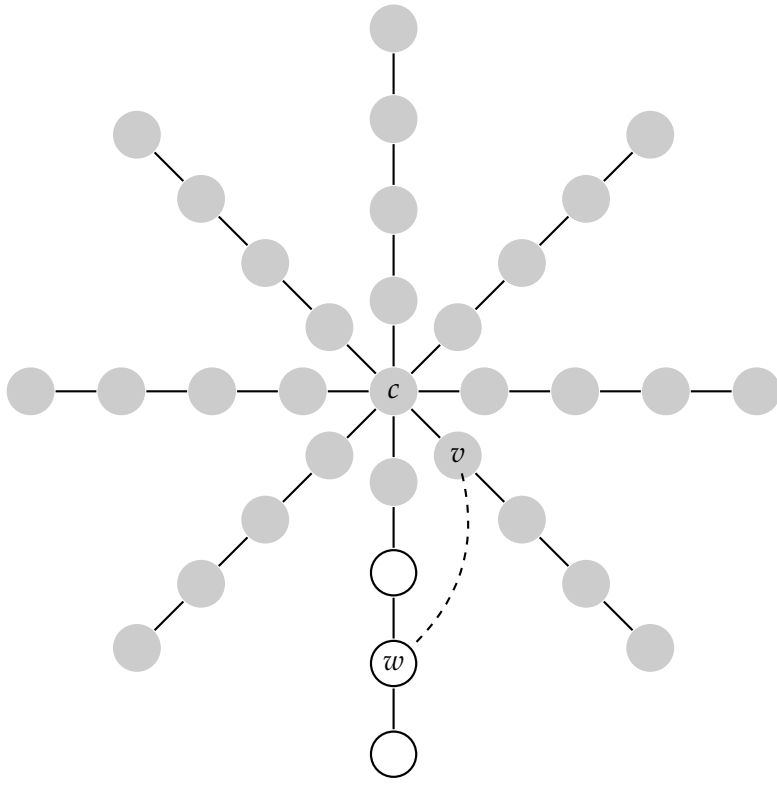


Figure 4.7. When the dashed edge is built, player v reduces distance only to those vertices drawn non-filled.

Now we aim for the construction of costly pairwise stable graphs. A *generalized star* with ray length ℓ and N rays is a graph on $n = N\ell + 1$ vertices, consisting of N paths of length ℓ (so each has $\ell + 1$ vertices) connected all together at one of their end vertices, making it a single vertex called the center vertex. Fig. 4.6 on the preceding page shows an example.

4.25 Proposition. *Let $\ell \in \mathbb{N}_{\geq 1}$ and $\alpha > 2\ell^2$. Then a generalized star with ray length ℓ and any number of rays N is pairwise stable. It has social cost at least $2(n-1)\alpha + (N-1)N\ell^2(\ell+1) = \Omega(n\alpha + n^2\ell)$.*

Proof. Clearly, no edge can be sold, since that would make the graph disconnected. Let c be the center vertex and $v, w \in V$ chosen so that

$d_v \leq d_w$, where $d_v := \mathrm{dist}(v, c)$ and $d_w := \mathrm{dist}(w, c)$. If $\{v, w\}$ is added, then v reduces her distance to at most ℓ vertices, since all these vertices are on w's ray, the center vertex excluded. Fig. 4.7 on the preceding page shows an example. The decrease per vertex is at most the former distance, which is at most 2ℓ, so the savings in total are at most $2\ell^2$, which is strictly smaller than α be assumption. Hence adding this edge would be an impairment to v. It follows that the graph is pairwise stable.

For computing the social cost, we use that the indirect cost of a tree T can be computed by summing over all edges, namely

$$I(T) = 2 \sum_{e \in E(T)} \nu(e)\,(n - \nu(e)) \ ,$$

where $\nu(e)$ denotes the number of vertices in that component of $T - e$ with a minimum number of vertices. We compute this sum ray-wise and get for the indirect cost:

$$\begin{aligned} 2N \sum_{i=1}^{\ell} i\,(n-i) &= 2N \left(n\frac{\ell\,(\ell+1)}{2} - \frac{\ell\,(\ell+1)\,(2\ell+1)}{6} \right) \\ &= N\ell\,(\ell+1) \left(n - \frac{2}{3}\ell - \frac{1}{3} \right) \\ &= N\ell\,(\ell+1) \left(N\ell + 1 - \frac{2}{3}\ell - \frac{1}{3} \right) \\ &\geq (N-1)\,N\ell^2\,(\ell+1) \ . \end{aligned}$$

□

4.26 Theorem. *In the bilateral sum-distance model, for $\varepsilon \in (0,1)$, $\alpha \geq 2+\varepsilon$ and $n-1$ being a multiple of $\lfloor \sqrt{\frac{\alpha-\varepsilon}{2}} \rfloor$, we have the following lower bound on the price of anarchy:*

(i) If $\alpha \leq n$, then $\rho(n, \alpha) = \Omega(\sqrt{\alpha})$.

(ii) If $\alpha \geq n$, then $\rho(n, \alpha) = \Omega(\frac{n}{\sqrt{\alpha}})$.

Proof. Since $\alpha \geq 1$, a star is optimal with social cost $\Theta(n\alpha + n^2)$ by Prop. 4.24(ii). Let $\ell := \lfloor \sqrt{\frac{\alpha-\varepsilon}{2}} \rfloor$, then $\ell \geq 1$ and $\ell^2 \leq \frac{\alpha-\varepsilon}{2} < \frac{\alpha}{2}$. Then by Prop. 4.25, any generalized star on n vertices with ray length ℓ (it has $N = \frac{n-1}{\ell}$ rays then) is pairwise stable with social cost $\Omega(n\alpha + n^2\ell)$.

It follows

$$\rho(n,\alpha) = \Omega\left(\frac{n\alpha + n^2\ell}{n\alpha + n^2}\right) = \Omega\left(\frac{\alpha + n\ell}{\alpha + n}\right) = \Omega\left(\frac{\alpha + n\sqrt{\alpha}}{\alpha + n}\right) \quad .$$

If $\alpha \leq n$, then this is $\Omega(\sqrt{\alpha})$. If $\alpha \geq n$, then this is $\Omega(\frac{n}{\sqrt{\alpha}})$. □

Why does this example work in the bilateral and not in the unilateral case? The unilateral case requires us to consider the exchange of edges: for a Nash equilibrium, we have to make sure that no player can improve her cost by selling some of her edges and building some new ones instead. We have seen in Fig. 4.4 on p. 105 how this can defy situations in which simply adding new edges is impossible without strictly increasing cost. The same would happen to any strategy profile S for which $G^{\mathsf{U}}(S)$ is a generalized star with ℓ and N large enough. But even without the option for exchange, no generalized star with $\ell, N \geq 2$ that is supposed to be a Nash equilibrium can provide a ratio of more than $O(1)$ to the optimum in the unilateral case. Consider a modification of Nash equilibrium such that a strategy profile is already considered an equilibrium if no player can improve her cost by either selling some edges or buying some edges, but not both. When a player at the end of a ray connects to the center vertex, she reduces distance to at least $(N-1)\,\ell$ vertices by an amount of $\ell - 1$ each. So it would be required that $\alpha > (N-1)\,\ell\,(\ell-1)$. It follows, using a ray-wise computation as in the proof of Prop. 4.25, that the social cost is at most $(n-1)\,\alpha + (N\ell)^2\,(\ell+1) = O(n\alpha + \alpha N\ell) = O(n\alpha)$, and so this does not yield a ratio to the optimum of more than $O(1)$.

4.4.3 Transformations

We point out two special cases in which a Nash equilibrium in the unilateral case can be transformed into a pairwise Nash equilibrium in the bilateral case. In the unilateral case, we call a Nash equilibrium S a *maximal Nash equilibrium* if $C_v(S + (v,w_1) + \ldots + (v,w_k), \alpha) > C_v(S,\alpha)$ for all $\{v,w_1\}, \ldots, \{v,w_k\} \notin E(S)$. That is, we exclude the possibility that a player can buy additional links so that the gain in her indirect cost and the additional building cost nullify each other.

4.27 Remark. In particular, a Nash equilibrium is maximal, if indirect cost has its minimum possible value for all players (which is 0 for most models). A Nash equilibrium is also maximal, if there exists $\varepsilon > 0$ such that it is still a Nash equilibrium for $\alpha - \varepsilon$ instead of α. Hence, if S is a Nash equilibrium for all $\alpha \geq f(n)$, for some function f, this implies that S is a maximal Nash equilibrium for all $\alpha > f(n)$.

Let S be a strategy profile. Define S^{B} by $S^{\mathsf{B}}_{vw} := \min\{1, S_{vw} + S_{wv}\}$ for all $v, w \in V$. Then $G^{\mathsf{U}}(S) = G^{\mathsf{B}}(S^{\mathsf{B}})$.

4.28 Proposition. *Let S be a maximal Nash equilibrium (in the unilateral case) with $G := G^{\mathsf{U}}(S)$ being a cycle. Then S^{B} is a pairwise Nash equilibrium in the bilateral case.*

Proof. By the definition of maximal Nash equilibrium, any additional link is an impairment for the buyer. So the premise of (4.3) is never true, i.e., all absent edges are justified.

New edges cannot be formed unilaterally. We are hence left to show that each edge is wanted by both endpoints, i.e., none of the endpoints can improve her individual cost by deleting the edge. Let v be the owner of an edge $\{v, w\}$ in the unilateral case. Since we have a Nash equilibrium there, v cannot improve her individual cost by selling this edge. Selling the edge means that v would be at the end of the path $G - \{v, w\}$. We note: it is worth or at least no impairment paying α for not being at the end of the path that results from G by deletion of one edge. Therefore, both of each two neighboring vertices maintain their requests in S^{B} for having an edge between them. □

4.29 Proposition. *Let S be a Nash equilibrium (in the unilateral case) with $G := G^{\mathsf{U}}(S)$ being a tree. Let the indirect cost assign ∞ to a disconnected graph. Then S^{B} is a pairwise Nash equilibrium in the bilateral case.*

Proof. As in the previous proposition, (4.3) follows from the properties of a maximal Nash equilibrium. So we are left to consider removals. Since the final graph is a tree, removal of any edge would make it disconnected, yielding indirect cost ∞. Hence no player wishes to remove an edge. □

4.5 Bibliographic Overview

We give a short bibliographic overview. It is by far not complete. In particular, the reader is encouraged to also consult the references in the cited publications.

4.5.1 Sum-Distance Model

The work of Fabrikant, Luthra, Maneva, Papadimitriou, and Shenker [41] from 2003 is to the best of my knowledge the first quantitative study of the price of anarchy in a model that fits into the framework considered here (Sec. 4.2). They considered the unilateral sum-distance model and proved a bound of $\max\{1, O(\sqrt{\alpha})\}$ on the price of anarchy in general, and an $O(1)$ bound for $\alpha > \frac{(n-1)\,n}{2}$; cf. Thm. 4.13. They conjectured that for $\alpha = \Omega(1)$, all non-transient Nash equilibria were trees – the Tree Conjecture. A Nash equilibrium is called *transient* when there exists a sequence of strategy changes in which each player changing her strategy maintains her individual cost, and finally a strategy profile is reached which is no Nash equilibrium anymore. The Tree Conjecture was based on the observation that all Nash equilibria constructed so far at that time, for $\alpha > 2$, were trees or transient ones (namely the Petersen graph for $\alpha \leq 4$). The Tree Conjecture was later, in 2006, disproved by Albers, Eilts, Even-Dar, Mansour, and Roditty [2] by showing that for each n_0, there exists a non-transient Nash equilibrium graph on $n \geq n_0$ vertices containing cycles, for any $1 < \alpha \leq \sqrt{n/2}$.

Corbo and Parkes [32] in 2005 considered the bilateral version of the sum-distance model. They showed an $O(\sqrt{\alpha})$ bound for $1 \leq \alpha < n^2$ on the price of anarchy. As noticed later in 2007 by Demaine et al. [38], the proof in fact yields $O(\min\{\sqrt{\alpha}, n/\sqrt{\alpha}\})$.

Albers et al. [2] in 2006 not only disproved the Tree Conjecture, but also improved the bounds on the price of anarchy for the unilateral sum-distance model: they gave constant upper bounds for $\alpha = O(\sqrt{n})$ and $\alpha \geq 12n\lceil\log n\rceil$, as well as an upper bound for any α of

$$15\ \left(1 + \left(\min\{\alpha^2/n,\ n^2/\alpha\}\right)^{1/3}\right)\ .$$

An $O(1)$ upper bound for $\alpha = O(\sqrt{n})$ was independently also proved

by Lin [57]. These bounds were again improved by Demaine, Hajiaghayi, Mahini, and Zadimoghaddam [38] in 2007. They showed a bound of $2^{O(\sqrt{\log n})}$ for any α and a constant bound for $\alpha = O(n^{1-\varepsilon})$ for any constant $\varepsilon > 0$. For the bilateral version, they proved the $O(\min\{\sqrt{\alpha}, n/\sqrt{\alpha}\})$ bound of Corbo and Parkes tight; we followed their proof for the lower bound in Sec. 4.4.2.

4.5.2 Equilibrium Concepts and Further Models

There is a vast body of literature on game-theoretic network formation, by far not limited to studies of the price of anarchy. A good starting point is the survey by Jackson [49] from 2004.

The way links are formed in the bilateral case follows a concept given by Myerson [67] in a different context. We quote [67, p. 228], emphasis added:

> Now consider **a link-formation process in which each player independently writes down a list of players with whom he wants to form a link**, and the payoff allocation is the fair allocation above for **the graph that contains a link for every pair of players that have named each other.**

Jackson and Wolinsky [50] in 1996 introduced the symmetric connections model and the equilibrium concept of pairwise stability. The symmetric connections model is best described using the notions of income and payoff, cf. p. 97. The income for player v is $\sum_{\substack{w \in V \\ w \neq v}} \delta^{\mathrm{dist}_{G(S)}(v,w)}$, where $\delta \in (0,1)$ is a parameter. Her payoff is income minus building cost. Note that we have an exponential dependence on distance. This models to some extent that each link has a probability of $1 - \delta$ for failure. We will elaborate on this in the next chapter.

Jackson and Wolinsky discussed several variations of pairwise stability, including what would later be known as pairwise Nash equilibrium. We quote [50, p. 67]:

> Another possible strengthening of the stability notion would allow for richer combinations of moves to threaten the stability of a network. Note that the basic stability notion we have considered requires only that a network be immune to one deviating

> action at a time. It is not required that a network be immune to more complicated deviations, such as a simultaneous severance of some existing links and an introduction of a new link by two players [..].

Watts [88] in 2001 studied the symmetric connections model with an extended equilibrium concept: a graph is considered stable if no player wishes to sell any link and if no two players wish to establish an additional link while deleting any number of their links. Calvó-Armengol and İlkiliç [23] and Corbo and Parkes [32] in 2005 discussed different equilibrium concepts and their relations: pairwise Nash equilibrium, pairwise stability, and proper equilibrium [67]. In [23], among other results, it was shown that the symmetric connections model has convex individual cost functions.

Bloch and Jackson [14] in 2007 introduced a model with transfers: each player v decides how much she is willing to pay for a link $\{v, w\}$ or how much she would demand the other endpoint w to pay for the link. If v offers at least as much as w demands, or vice versa, the link $\{v, w\}$ is established in the final graph. Appropriate equilibrium concepts were introduced and discussed. Bloch and Jackson also compared pairwise stability, pairwise Nash equilibrium, and their transfer model in a separate publication [13].

Bala and Goyal [7] in 2000 and in a unilateral setting studied a model where players wish to be connected by a path to as many other players as possible, but path lengths are unimportant. They also considered a unilateral version of the symmetric connections model. In another publication [8] in the same year, they extended the first model by allowing each link to fail with a probability $1 - p$. Haller and Sarangi [46, 82] in 2003 extended this model again by allowing each link $\{v, w\}$ to fail with its own probability $1 - p_{vw}$. We will elaborate on this in Sec. 5.2 in the next chapter.

Anshelevich, Dasgupta, Tardos, and Wexler [5] in 2003 studied the price of anarchy and algorithmic aspects of a model in which each player has a set of terminals and aims to construct a network which connects her terminals. For a related model, Anshelevich, Dasgupta, Kleinberg, Tardos, Wexler, and Roughgarden [4] in 2004 studied the price of stability. Also in 2004, Christin and Chuang [26] studied a

model for network formation with an extended cost function modeling peer-to-peer networks, and Christin, Grossklags, and Chuang [27] looked at it under the aspect of different game-theoretic principles.

Chun, Fonseca, Stoica, and Kubiatowicz [29] in 2004 experimentally studied an extended version of the sum-distance model.

Johari, Mannor, and Tsitsiklis [51] in 2006 studied a model in which each vertex wishes to send a given amount of traffic to some of the other vertices, and only cares whether the traffic eventually arrives at the destination. There is a handling cost at each vertex, which is proportional to the amount of traffic through that vertex.

Moscibroda, Schmid, and Wattenhofer [66] in 2006 studied the price of anarchy in a variation of the sum-distance model where the distance between two vertices is generalized, that is, it may be given by any metric. The cost function uses the stretch, that is the actual distance in the constructed graph divided by the distance that a direct connection would provide. Halevi and Mansour [45] in 2007 studied the price of anarchy in the sum-distance model under the generalization that each player has a list of "friends", that is, a list of other vertices and she is only interested in her distance to those. Demaine et al. in [38] in 2007 also considered the max-distance model: indirect cost for v is $\max_{w \in V} \mathrm{dist}(v, w)$. Upper bounds were shown for the unilateral case and tight bounds for the bilateral case. Brandes, Hoefer, and Nick [20] in 2008 studied a variant of the sum-distance model assigning a finite distance to pairs of disconnected players, allowing for disconnected equilibria. They proved structural properties and bounds on the price of anarchy. Laoutaris, Poplawski, Rajaraman, Sundaram, and Teng [56] in 2008 considered a variant of the sum-distance model with player-dependent link costs, lengths, and preferences $w(u, v)$ expressing the importance of player u for having a good connection to player v, and finally a budget for each player limiting the number of links that this player can build. They considered existence of equilibria and proved bounds on the price of anarchy and stability. Baumann and Stiller [10] in 2008 considered the price of anarchy in the symmetric connections model. Demaine et al. [39] in 2009 studied the price of anarchy in a cooperative variant of the sum-distance model. They also looked at the case that links can only be formed for certain pairs of vertices, in other words the underlying "host" graph needs not to be a complete one.

Chapter 5

Distributed Network Formation Against an Adversary

As pointed out by Fabrikant et al. [41], the sum of distances does by far not capture all potentially relevant aspects of a graph; in particular, it misses congestion and robustness properties. In this chapter we are concerned with robustness properties of the final graph. Such has already been addressed in specific forms before, but has to the best of my knowledge not yet been studied theoretically in a model like this. The model is introduced in Sec. 5.1. We review previous work which addresses robustness and explain features and differences of the various models in Sec. 5.2. The remaining sections analyze our model with a focus on the price of anarchy. The most interesting upper bounds are for the unilateral case, presented in Sec. 5.4 and Sec. 5.5. The bilateral case is also interesting, as we show in Sec. 5.6: it exhibits a much higher price of anarchy for a certain subclass than we have in the unilateral case. It so raises the question about how our model behaves in alternative equilibrium concepts that are *between* the unilateral and bilateral case, e.g., models with transfers like the one recently introduced by Bloch and Jackson [14]; we leave this for future work. We summarize our new results in Sec. 5.7.

5.1 Model

We use the framework from Sec. 4.2. For the final graph, we consider each of the two definitions from Def. 4.1, i.e., the unilateral and the bilateral case. The indirect cost is chosen to express robustness properties of the final graph in the following way. Imagine that after the network is built, an adversary deletes one link of his choice. Players are not allowed to react to the adversary's decision by replacing the deleted link immediately – they have instead to cope with the damaged network and use it to perform their tasks as good as possible at least for some period of time. So players have to think ahead and build the network as robust as it seems appropriate given the link cost α. The adversary is modeled by a random experiment, i.e., we specify a probability distribution on the links of the final graph and let the adversary pick its target according to this distribution. So players know in advance that exactly one link will be deleted, but in general they do not know which one it will be. Knowing the adversary's probability distribution, they can, however, make statements in terms of expectation. We define the objective of each player to stay connected to as many other vertices as possible, in expectation.

Let a strategy profile S be given and $\Pr_{G(S)}$ be a probability distribution on the links $E(S)$ of the final graph. It is $\Pr_{G(S)}(\{e\})$ the probability that the adversary will delete link e. Note that we allow the distribution to depend on the final graph. If $G(S)$ is connected, we define the indirect cost of player v to be the expected number of vertices to which v will lose connection when the adversary strikes, given $\Pr_{G(S)}$. If $G(S)$ is not connected, we define the indirect cost to be ∞ for all players. In the following, we will use the term *disconnection cost* instead of "indirect cost" for the sake of clearness. Neither an optimum nor an equilibrium is disconnected, and so we will always assume that $G(S)$ is connected.

Define the *relevance* $\mathrm{rel}_{G(S)}(e, v)$ of a link e for player v to be the number of vertices that can, starting at v, *only* be reached via e. In particular, $\mathrm{rel}_{G(S)}(e, v) = 0$ for all $v \in V$ if e is on a cycle, or – equivalently – a non-bridge. If link e is deleted (by the adversary), then player v loses connection to exactly $\mathrm{rel}_{G(S)}(e, v)$ other vertices. For each $v \in V$, denote

the sum of relevances $R_{G(S)}(v) \coloneqq \sum_{e \in E} \mathrm{rel}(e, v)$. For each $e \in E(S)$, denote $\mathrm{sep}_{G(S)}(e) \coloneqq \sum_{v \in V} \mathrm{rel}_{G(S)}(e, v)$ the number of separated vertex pairs when e is deleted; it is $\mathrm{sep}_{G(S)}(e) = 2\,\nu_{G(S)}(e)\,(n - \nu_{G(S)}(e))$. We so have the individual and total indirect cost, respectively:

$$\begin{aligned} I_v(G(S)) &\coloneqq \sum_{e \in E(S)} \Pr_{G(S)}(\{e\})\,\mathrm{rel}_{G(S)}(e, v) && \text{for } v \in V, \\ I(G(S)) &= \sum_{v \in V} \sum_{e \in E(S)} \Pr_{G(S)}(\{e\})\,\mathrm{rel}_{G(S)}(e, v) \\ &= \sum_{e \in E(S)} \Pr_{G(S)}(\{e\})\,\mathrm{sep}_{G(S)}(e) \ . \end{aligned}$$

5.1.1 Simplified Notation

As in the previous chapter, we use a closely graph-related notation. We roughly repeat the discussion from Sec. 4.2.2 here, filling in all concrete definitions of our model, like we did in Sec. 4.3 for the sum-distance model.

Most of the time we will not work with strategy profiles but with the final graph directly and denote it simply by $G = (V, E)$ and set $m \coloneqq |E|$. All subscripts "$G(S)$" are omitted; everything relates to the graph G. So we write "$\mathrm{rel}(e, v)$" instead of "$\mathrm{rel}_{G(S)}(e, v)$". The social cost only depends on G (and α), and so we write "$C(G, \alpha)$", but omit "G" otherwise. The social cost for the unilateral case is

$$\begin{aligned} C^{\mathsf{U}}(G, \alpha) &= m\,\alpha + \sum_{v \in V} \sum_{e \in E} \Pr(\{e\})\,\mathrm{rel}(e, v) \\ &= m\,\alpha + \sum_{e \in E} \Pr(\{e\})\,\mathrm{sep}(e) \ . \end{aligned}$$

For the bilateral case, the social cost is

$$\begin{aligned} C^{\mathsf{B}}(G, \alpha) &= 2m\,\alpha + \sum_{v \in V} \sum_{e \in E} \Pr(\{e\})\,\mathrm{rel}(e, v) \\ &= 2m\,\alpha + \sum_{e \in E} \Pr(\{e\})\,\mathrm{sep}(e) \ . \end{aligned}$$

The only difference is the factor 2 in the building cost.

In the unilateral case, it sometimes is important which player pays for which edge. Then we use a directed version of G to express that, denoted $\vec{G}$, i.e., we choose $\vec{G}$ such that $\vec{G} = \vec{G}(S)$, where S is the strategy profile we wish to express. Then the number of edges for which v has to pay is her out-degree $\deg^{\text{out}}(v)$, with respect to $\vec{G}$. We can then express the individual cost of player v in only graph-theoretic terms:

$$C_v^{\mathsf{U}}(\vec{G}, \alpha) = \deg^{\text{out}}(v)\,\alpha + \sum_{e \in E} \Pr(\{e\})\,\text{rel}(e, v) \ .$$

For the bilateral case, it is even easier. We only have to consider the degree $\deg(v)$ with respect to G, and so denote the individual cost dependent on G:

$$C_v^{\mathsf{B}}(G, \alpha) = \deg(v)\,\alpha + \sum_{e \in E} \Pr(\{e\})\,\text{rel}(e, v) \ .$$

Since connectivity under deletion of an edge is of importance, one may ask whether it would be wise to allow multigraphs instead of graphs now. This question will have been answered by the end of this chapter: none of our results becomes false when we allow multigraphs. In places where this is not obvious, explanations are given. Hence there would be no justification to go through the unpleasantness of adapting all formalism to multigraphs.

5.1.2 Illustration

We provide an illustrative interpretation of the individual cost. Consider that each vertex v has a facility to efficiently produce a commodity of type t_v, called the *native commodity* of v. All types $(t_w)_{w \in V}$ are distinct, and each vertex requires commodities of each type. For vertex v, commodities of type t_w, $w \neq v$, are called *foreign commodities*. As long as v is connected to all other vertices via paths in the final graph, v receives all foreign commodities from other vertices. Each vertex v is also able to produce foreign commodities locally, i.e., directly at v and not requiring any network. However, local production of foreign commodities is less efficient and hence more costly than production of native ones. If a vertex was disconnected from only one other vertex

for a longer period of time, say, for one time unit, the extra expenses would exceed her budget. This is expressed by the indirect cost being ∞ for a disconnected final graph. Construction and maintenance of each link costs α per time unit for the vertex which built the link. The adversary destroys one link per time unit. Then, some vertex pairs may be disconnected, and so, some foreign commodities have to be produced locally. Fortunately, the link can be repaired in a period of time that is relatively short compared to one time unit, so the budget is not necessarily exceeded. Let the extra cost for local production of foreign commodities while the link is being repaired be 1 for each vertex and each foreign commodity. Then the disconnection cost of v is the expected extra cost per time unit.

5.2 Previous Work and Comparison

Chun, Fonseca, Stoica, and Kubiatowicz [29] experimentally study an extended version of the sum-distance model. They also address robustness. To simulate failures, they remove some vertices randomly. To simulate attacks, they remove vertices starting with those having highest degree.

The symmetric connections model of Jackson and Wolinsky [50] can also be interpreted from a robustness point-of-view. Recall that in the symmetric connections model there is a parameter $\delta \in (0,1)$, and the payoff $\pi_v(S,\alpha)$ for player v under strategy profile S is defined

$$\pi_v(S,\alpha) := \sum_{\substack{w \in V \\ w \neq v}} \delta^{\mathrm{dist}_{G(S)}(v,w)} - \deg_{G(S)}(v)\,\alpha \ .$$

An interpretation is that v receives one unit of income from each other vertex w along a shortest path between v and w. However, each link has a probability $1-\delta$ of failure, so the expected income from w is the probability that none of the $\mathrm{dist}_{G(S)}(v,w)$ links fails, which is $\delta^{\mathrm{dist}_{G(S)}(v,w)}$ if we assume stochastic independence of failures. In the bilateral case, Baumann and Stiller [10] give an expression for the exact price of anarchy for $\alpha \in (\delta-\delta^2, \delta-\delta^3)$, which implies an $O(1)$ bound. The price of anarchy is 1 for $\alpha < \delta - \delta^2$, following from [50]. The price of anarchy in the range $\alpha > \delta - \delta^3$ is not fully understood yet.

The symmetric connections model is different from ours in many respects:

– All links have the same probability of failure. In our model, links can have different probabilities, and these may even depend on the final graph.[1]

– The failure of a link e and the failure of a link f are independent events for $e \neq f$, at least along the concerned paths. In our model, the failures of e and f are mutually exclusive events.

– Alternative paths are not considered; it is assumed that routing happens along a specific shortest path that is fixed before the random experiment that models the link failures is conducted. In our model, *all* paths are considered. However, we do not consider path lengths.

Bala and Goyal [8] study a variation of the symmetric connections model, which is closer to ours. In their model, each vertex receives an amount of 1 from each vertex it is connected to via some path. Each link has a probability $1 - p$ of failure, $p \in [0, 1]$ being the same for all links and independent of the final graph. Failures of two distinct links are stochastically independent. The income of a vertex v is the expected number of vertices to which v is connected via a path. Unilateral link formation is used. They consider structural properties of optima and Nash equilibria, in particular pointing out cases where Nash equilibria are "super-connected", i.e., connected and not containing bridges. They also show that for some regions of parameters, there exist Nash equilibria that are also optima (i.e., they show a price of stability of 1 for these regions).

Haller and Sarangi [46, 82] study an extension of the model of Bala and Goyal [8]. In their model, each link $\{v, w\}$ may fail with its *own* probability $1 - p_{vw}$. They also consider structural properties of optima and Nash equilibria as well as relations of optima and Nash equilibria, including the price of stability similar to [8]. Like the symmetric connections model, their model shows several differences to ours:

– The failure probability of each link $\{v, w\}$ is $1 - p_{vw}$, independent of

[1] However, our analysis will be restricted to two specific cases: one in which the adversary picks a link uniformly at random and another in which he picks a link that causes maximum overall damage.

the final graph.[2] In our model, failure probabilities depend on the final graph.

– Failures of two different links are stochastically independent. In our model, they are mutually exclusive events. (This difference is exactly as between the symmetric connections model and ours.)

5.3 General Bounds and the Bridge Tree

We turn to our model now. Throughout this chapter we assume $n \geq 3$. Without further knowledge on $\Pr_{G(\cdot)}(\cdot)$, and also without specifying whether we use G^{U} or G^{B}, or which equilibrium concept, we can give two simple bounds on the price of anarchy.

5.1 Proposition. *Fix any link formation rule and equilibrium concept.*

(i) The price of anarchy is $O(n + \frac{n}{\alpha})$.

(ii) If $m = O(n)$ *for all equilibria, and if* $\alpha = \Omega(n)$, *then the price of anarchy is* $O(1)$.

Proof. Since $m = O(n^2)$ and $\operatorname{sep}(e) = O(n^2)$ for all e, we have

$$C(G, \alpha) = O\big(m\alpha + n^2 \sum_{e \in E} \Pr(\{e\})\big) = O(m\alpha + n^2) = O(n^2\alpha + n^2)$$

in general. Since an optimum is connected, the optimal cost is $\Omega(n\alpha)$. Both bounds on the price of anarchy follow. □

Later, in Sec. 5.5.2, when bounding the price of anarchy by $O(1)$, we will show $m = O(n)$ first, and then by the previous proposition we are allowed to restrict to $\alpha \leq n$.

In order to analyze equilibria, it will be helpful in several places to consider a variation of the block graph,[3] which we call the *bridge tree*. Its definition requires some preparation. If $W \subseteq V$ is maximal

[2] Haller and Sarangi also briefly discuss failure probabilities depending on the final graph. They consider an example where for non-increasing functions $f_v(\cdot)$ and parameters P_{vw} the probabilities are defined $p_{vw}(S) := f_v(\deg_{G(S)}(v))\, f_w(\deg_{G(S)}(w))\, P_{vw}$ if v and w have a link between them, and 0 otherwise.

[3] See, e.g., [40, p. 56] for the definition of the block graph.

under the condition that the induced subgraph $G[W]$ is connected and does not contain any bridges of $G[W]$, we call W a *bridgeless connected component*, abbreviated "BCC".

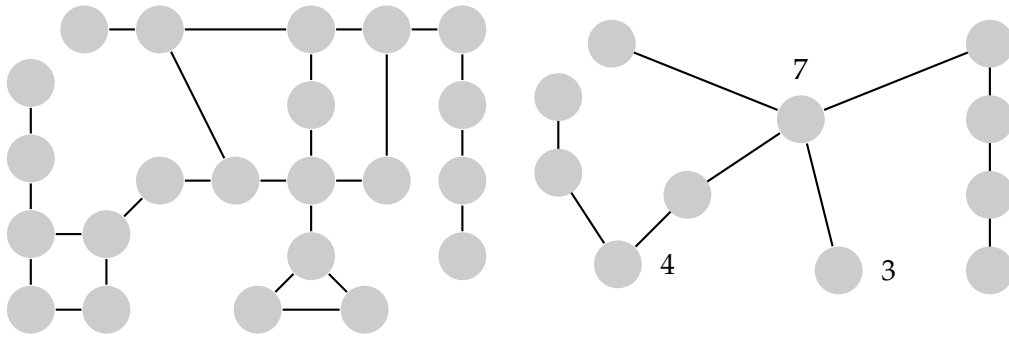
5.2 Proposition. *We get exactly the same* BCCs *when we request that* $G[W]$ *does not contain any bridges of* G *in the above definition.*

Proof. Let W be maximal under the condition of $G[W]$ being connected and not containing any bridges of $G[W]$, i.e., we follow the original definition given above. Clearly, $G[W]$ does not contain any bridges of G, since if removal of some edge disconnects G, then it also disconnects $G[W]$ if the endpoints of this edge are in W. We choose $U \supseteq W$ maximal under the condition that $G[U]$ is connected and $G[U]$ does not contain any bridges of G. Suppose $U \neq W$. Then $G[U]$ contains a bridge e of $G[U]$. Since this is no bridge of G, it is located on a cycle C. Then $V(C) \not\subseteq U$, since e is a bridge of $G[U]$. But $G[U \cup V(C)]$ would still be connected and would contain no bridge of G. This contradicts the maximality of U.

Now let W be maximal under the condition of $G[W]$ being connected and not containing any bridges of G. If $G[W]$ contained a bridge e of $G[W]$ (but not of G), we could use the cycle-argument from before to augment W and have a contradiction to its maximality. Suppose there is $U \supsetneq W$ such that $G[U]$ is connected and $G[U]$ does not contain any bridges of $G[U]$. Then $G[U]$ contains a bridge of G. As noted earlier, this is also a bridge of $G[U]$, a contradiction. □

What we call "BCC" is sometimes called "block" in the literature, and what we call "bridge tree" is then called "bridge-block tree". We refrain from using the term "block" here, since it usually is related to *vertex*-connectivity; see, e.g., [40, p. 55].

Every vertex is contained in exactly one BCC. If W is a BCC, we have to remove at least 2 edges from $G[W]$ in order to make it disconnected. A graph from which we have to remove at least 2 edges to make it disconnected is also called being "2-edge-connected" in common terminology, provided that it has more than 1 vertices; see, e.g., [40, p. 12].



a. A graph G.

b. The corresponding bridge tree $\widetilde{G}$.

Figure 5.1. Bridge tree construction. Vertices representing BCCs of more than 1 vertices have their number of vertices attached, here 4, 7, and 3, respectively.

Now we introduce the bridge tree. It is the graph $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ defined by:

$$\widetilde{V} := \{B \subseteq V;\ B \text{ is a BCC}\}\ ,$$
$$\widetilde{E} := \{\{B, B'\};\ B, B' \in \widetilde{V}, \exists v \in B, w \in B' : \{v, w\} \in E\}\ .$$

Then $\widetilde{G}$ is a tree (assuming G is connected). By Prop. 5.2, there is a $1:1$ mapping between the edges of $\widetilde{G}$ and the bridges of G. We make the following special convention concerning the bridge tree:

Convention. Whenever we speak of the number of vertices in a subgraph T of the bridge tree, we count $|B|$ for each vertex $B \in V(T)$.

In other words, we count the vertices that would be there if we expanded T back to its corresponding subgraph of G. Fig. 5.1 shows an example. Since each vertex of G is in exactly one BCC, counting in this way for $\widetilde{V}$ yields the number of vertices in G, i.e., n.

On several occasions, when considering the effect of building additional edges, we treat vertices of the bridge tree as players. This is justified since edges inside BCCs have relevance 0. Hence for a strategy profile S and $B, B' \in \widetilde{V}$ the effect in disconnection cost of a new edge between a player from B and a player of B' is specific to the pair $\{B, B'\}$ and not to the particular players.

For a path P in G, let $\widetilde{P}$ be its contracted counterpart in $\widetilde{G}$, i.e., we replace in P each maximal sequence of vertices from the same BCC $B \in \widetilde{V}$ with B. Then the length $|\widetilde{P}|$ of $\widetilde{P}$ is the number of bridges in P. For each pair $v, w \in V$ denote $P(v,w)$ an arbitrary shortest path from v to w; and $\mathcal{P}(v) \coloneqq \{P(v,w);\ w \in V\}$. The bridge tree helps bounding the disconnection cost. We conclude this section with a preparation for this. For each $v \in V$ and $e \in E$ we easily observe:

$$\mathrm{rel}(e,v) = \begin{cases} 0 & \text{if } e \text{ is a non-bridge} \\ |\{P \in \mathcal{P}(v);\ e \in E(P)\}| & \text{if } e \text{ is a bridge.} \end{cases} \tag{5.1}$$

5.3 Lemma. *For each $v \in V$ we have $R(v) \leq (n-1)\,\mathrm{diam}(\widetilde{G})$.*

Proof. The idea is that it is the same to count for each edge the number of paths that cross this edge as to count for each path the number of its edges. Fix $v \in V$. We have

$$\begin{aligned} R(v) \overset{\text{def}}{=} \sum_{e \in E} \mathrm{rel}(e,v) &= \sum_{\substack{e \in E \\ e \text{ is a bridge}}} |\{P \in \mathcal{P}(v);\ e \in E(P)\}| \\ &= \sum_{P \in \mathcal{P}(v)} |\{e \in E(P);\ e \text{ is a bridge}\}| \\ &= \sum_{P \in \mathcal{P}(v)} |\widetilde{P}| \\ &\leq (n-1)\,\mathrm{diam}(\widetilde{G})\ . \end{aligned}$$

The last estimation is true since the bridge tree is a tree and so every path is a shortest path. □

5.4 Simple-Minded Adversary

We work in the unilateral case and consider an adversary that picks one edge uniformly at random, i.e., $\Pr(\{e\}) = \frac{1}{m}$ for each $e \in E$. Then we have the individual and social cost, respectively:

$$C_v(\vec{G}, \alpha) = \deg^{\text{out}}(v)\,\alpha + \frac{1}{m} \sum_{e \in E} \mathrm{rel}(e,v) \quad \text{for } v \in V,$$

$$C(G,\alpha) = m\,\alpha + \frac{1}{m}\sum_{v\in V}\sum_{e\in E}\mathrm{rel}(e,v) = m\,\alpha + \frac{1}{m}\sum_{e\in E}\mathrm{sep}(e)\ .$$

As an example, we consider the path. This example shows that the social cost can very well be of order $\Omega(n\alpha + n^2)$.

5.4 Proposition. *The social cost of the path is* $(n-1)\,\alpha + \frac{1}{3}n\,(n+1) = \Theta(n\alpha + n^2)$.

Proof. We have the social cost of the path:

$$\begin{aligned}
&m\,\alpha + \frac{1}{m}\sum_{v\in V}\sum_{e\in E}\mathrm{rel}(e,v) = (n-1)\,\alpha + \frac{1}{n-1}\sum_{e\in E}\sum_{v\in V}\mathrm{rel}(e,v)\\
&= (n-1)\,\alpha + \frac{1}{n-1}\sum_{e\in E}\mathrm{sep}(e)\\
&= (n-1)\,\alpha + \frac{1}{n-1}\sum_{e\in E}2\,\nu(e)\,(n-\nu(e))\\
&= (n-1)\,\alpha + \frac{2}{n-1}\sum_{k=1}^{n-1}k\,(n-k)\\
&= (n-1)\,\alpha + \frac{2}{n-1}\left(n\frac{(n-1)\,n}{2} - \frac{(n-1)\,n\,(2n-1)}{6}\right)\\
&= (n-1)\,\alpha + \frac{1}{3}n\,(n+1) = \Theta(n\alpha + n^2)\ .
\end{aligned}$$

□

We now show how optima look like in our model. Then we will investigate what kind of cycles can occur in Nash equilibria and give some concrete Nash equilibrium graphs; this shows existence of Nash equilibria for all n and α, and it proves a lower bound on the price of anarchy and an upper bound on the price of stability. Then we upper-bound the social cost of Nash equilibria, which finally leads to our main result, the $O(1)$ bound on the price of anarchy.

5.4.1 Optima, Equilibria, and the Price of Stability

5.5 Proposition. *An optimum has cost* $\Theta(n\alpha)$. *More precisely:*

(i) If $\alpha \le 2\,(n-1)$, *the cycle is an optimum; it has cost* $n\alpha$.
If $\alpha < 2\,(n-1)$, *then the cycle is the only optimum.*

(ii) If $\alpha \geq 2(n-1)$, a star is an optimum; it has cost $(n-1)(\alpha+2)$. If $\alpha > 2(n-1)$, then stars are the only optima.

Proof. An optimum can only be the cycle or a tree. For, any graph containing a cycle has already the cost $n\alpha$ of the cycle, and the cycle has optimal disconnection cost. So an optimum is either the cycle, or it is cycle-free. Let T be any tree. We have

$$\begin{aligned}\sum_{e \in E(T)} \mathrm{sep}_T(e) &= 2 \sum_{e \in E(T)} \nu(e)\,(n - \nu(e)) \\ &\geq 2 \sum_{e \in E(T)} 1\,(n-1) = 2\,(n-1)^2 \ .\end{aligned}$$

Hence the cost of a tree is at least

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

The cost of the cycle is $n\alpha$. So if $\alpha \leq 2(n-1)$, the cycle is better or as good as any tree, so it is an optimum. If $\alpha \geq 2(n-1)$, then we look for a good tree. A star has cost $(n-1)(\alpha+2)$, which matches the lower bound given above, and is hence optimal. □

The following proposition is purely graph-theoretic and will be used here and later when we study convexity of cost functions.

5.6 Proposition. *Let $G = (V, E)$ be a graph and $e = \{v, w\} \in E$ a non-bridge. Let C be* any *cycle with $e \in E(C)$. Then all bridges of $G - e$ that are non-bridges in G, are in $E(C)$.*

Proof. Let f be a non-bridge in G and a bridge in $G - e$. Then $G - e$ consists of two subgraphs G_1 and G_2 that are connected only by f. Since f was no bridge before e was removed, e must also connect G_1 with G_2. Moreover, there are no other edges between G_1 and G_2. It follows that any cycle that contains e also contains f. □

Now we estimate the benefit for a player of building or selling a particular edge. Fix a player $v \in V$ and recall $R(v) = \sum_{e \in E} \mathrm{rel}(e, v)$.

5.7 Proposition. *We have $R(v) \leq \frac{n(n-1)}{2}$.*

Proof. We repeat the counting argument from the proof of Lem. 5.3:

$$R(v) \overset{\text{def}}{=} \sum_{e\in E} \operatorname{rel}(e,v) = \sum_{\substack{e\in E\\ e \text{ is a bridge}}} |\{P \in \mathcal{P}(v);\ e \in E(P)\}|$$
$$= \sum_{P\in\mathcal{P}(v)} |\{e \in E(P);\ e \text{ is a bridge}\}| \le \sum_{P\in\mathcal{P}(v)} |E(P)| = \sum_{w\in V} \operatorname{dist}(v,w)\ .$$

This is maximal if G is a path with v at its end; then $R(v) = \frac{n\,(n-1)}{2}$. □

Let $R := R(v)$ and let R' be the same quantity when an additional edge e is built by v. By the previous proposition, we have $R, R' \le \frac{n\,(n-1)}{2}$. The benefit in disconnection cost of building this edge for player v is $\frac{1}{m}R - \frac{1}{m+1}R'$. Due to the change in denominators from "m" to "$m+1$" this expression looks somewhat unhandily. Yet, we can give good bounds incorporating the change in relevances, $\Delta R := R - R' \ge 0$, with one denominator. We can do something similar for the case when the player *sells* an edge, where we put $\Delta R := R' - R \ge 0$.

5.8 Proposition.

(i) If a player builds an additional edge and the sum of her relevances decreases by ΔR, then her improvement in disconnection cost is at least $\frac{1}{m+1}\Delta R$ and at most $\frac{1}{2} + \frac{1}{m+1}\Delta R \le \frac{n}{2}$.

(ii) If a player sells a non-bridge and the sum of her relevances increases by ΔR, then her impairment in disconnection cost is at least $\frac{1}{m}\Delta R$ and at most $\frac{1}{2} + \frac{1}{m}\Delta R \le \frac{n}{2}$.

Proof. (i) We have

$$\begin{aligned}
&\frac{1}{m}R - \frac{1}{m+1}R' = \frac{1}{m}R - \frac{1}{m+1}(R + (R' - R))\\
&= \left(\frac{1}{m} - \frac{1}{m+1}\right)R + \frac{1}{m+1}\Delta R\\
&= \frac{1}{m\,(m+1)}R + \frac{1}{m+1}\Delta R \quad \begin{cases} \le \frac{1}{2} + \frac{1}{m+1}\Delta R \le \frac{n}{2} \\ \ge \frac{1}{m+1}\Delta R \end{cases}.
\end{aligned}$$

We used $\Delta R \le \frac{n\,(n-1)}{2}$ and $n - 1 \le m$ for the upper bound. (ii) is proved alike, using $n \le m$ since the graph contains a cycle. □

5.9 Proposition.

(i) If a player builds an edge creating a cycle of length ℓ, the improvement in disconnection cost is at most $\frac{1}{2} + \frac{1}{m+1}(\ell-1)(n-\frac{\ell}{2})$.

(ii) If a player sells an edge destroying a cycle of length ℓ, the impairment in disconnection cost is at most $\frac{1}{2} + \frac{1}{m}(\ell-1)(n-\frac{\ell}{2})$.

Proof. (i) The restriction to any single cycle is admissible since any new cycle contains all the edges that become non-bridges due to the new edge; this can be seen by considering the bridge tree. Let $C = (v, e_1, v_1, \ldots, v_{\ell-1}, e_\ell, v)$ be a new cycle, created by the new edge e_ℓ bought by v. In the best case, i.e., in case of maximal improvement, without the additional edge, $n-1$ vertices are reached from v only through e_1, $n-2$ through the next edge, and so on; edge $e_{\ell-1}$ is relevant for $(n-(\ell-1))$ vertices. It follows $\Delta R \leq \sum_{k=1}^{\ell-1}(n-k) = (\ell-1)\,n - \sum_{k=1}^{\ell-1} k = (\ell-1)\,n - \frac{(\ell-1)\,\ell}{2} = (\ell-1)\,(n-\frac{\ell}{2})$. The statement follows with Prop. 5.8(i).

(ii) The restriction to any single cycle is admissible by Prop. 5.6. The rest is the same calculation as for (i). □

5.10 Proposition. *Let $\ell < \alpha + \frac{1}{2}$.*

(i) If a player builds an additional edge creating a cycle of length ℓ, she suffers an impairment in her cost.

(ii) If a player sells an edge destroying a cycle of length ℓ, she experiences an improvement in her cost.

Proof. (i) By Prop. 5.9(i), the player suffers an impairment in her cost if

$$\alpha > \frac{1}{2} + \frac{1}{m+1}(\ell-1)\left(n-\frac{\ell}{2}\right) \quad .$$

Since $m \geq n-1$, this is the case if $\alpha > \frac{1}{2} + \frac{1}{n}(\ell-1)(n-\frac{\ell}{2})$, which is the same as $n(\alpha+\frac{1}{2}) > \ell(n-\frac{\ell}{2}+\frac{1}{2})$. Since $\ell \geq 3 \geq 1$, this is the case if $n(\alpha+\frac{1}{2}) > \ell n$.

We show (ii) in almost exactly the same way, using Prop. 5.9(ii) and that $m \geq n$, since the original graph contains a cycle. □

It follows a structural result:

5.11 Corollary. *No Nash equilibrium contains cycles shorter than* $\alpha + \frac{1}{2}$.

Nash equilibria always exist:

5.12 Proposition.

(i) If $\alpha \leq \frac{n}{2}$, *then the cycle with all edges pointing in the same direction (either all clockwise or all counter-clockwise) is a Nash equilibrium.*

(ii) If $\alpha \geq \frac{3}{2}$, *then a star with all edges pointing outwards is a Nash equilibrium.*

Proof. (i) The cycle has minimum disconnection cost, namely 0, so there is no incentive to buy any more edges. If a player sells her one edge e, we have $\Delta R = \frac{n(n-1)}{2}$, and since the new graph has $n-1$ edges, this means an impairment in disconnection cost of $\frac{n}{2} \geq \alpha$. Hence the player has no incentive to sell the edge. There is also no incentive in building one or more different edges instead of e, since that cannot reduce the disconnection cost to the original value of 0.

(ii) Since all edges point outward, the center is the only vertex that could sell an edge, but this would make the graph disconnected. The center cannot buy additional edges. An outer vertex cannot sell edges. We have to check whether it is beneficial for an outer vertex to buy additional edges. Building one or more edges induces per edge $\Delta R \leq n - 1 + 1 = n$, and so by Prop. 5.8(i) an improvement per new edge of at most $\frac{1}{2} + \frac{1}{n}n = \frac{3}{2} \leq \alpha$.

If multiple edges were allowed, also the center vertex could build additional edges. They would induce $\Delta R = 1$ each, and so an improvement of at most $\frac{1}{2} + \frac{1}{n} \leq \frac{3}{2} \leq \alpha$. □

5.13 Remark. By Rem. 4.27, in the previous proposition, the cycle is a maximal Nash equilibrium and the star is a maximal Nash equilibrium for $\alpha > \frac{3}{2}$. □

Knowing optima and examples for Nash equilibria, we can show:

5.14 Proposition. *We can choose* α *so that* $\rho(n, \alpha) \geq 2 + \frac{1}{3} - o(1)$ *for* $n \to \infty$.

Proof. Let $\alpha := \frac{3}{2}$ and n be large enough so that the cycle is optimal. Then a star with all edges pointing outward is a Nash equilibrium with social cost $(\alpha+2)(n-1) = \alpha n + 2n - (\alpha+2)$. Dividing that by $n\alpha$, the cost of the cycle, yields a lower bound of $1 + \frac{2}{\alpha} - o(1) = 1 + \frac{4}{3} - o(1)$ on the price of anarchy. □

5.15 Theorem. *The price of stability is* $1 + o(1)$ *for* $n \to \infty$.

Proof. For $\alpha \leq \frac{n}{2}$ the cycle is a Nash equilibrium as well as an optimum, and so the price of stability is 1. For $\alpha \geq 2(n-1)$ a star is a Nash equilibrium as well as an optimum, and so the price of stability is 1. For $\frac{n}{2} \leq \alpha \leq 2(n-1)$, a star is a Nash equilibrium and the cycle is an optimum. The price of stability so is upper-bounded by $1 + \frac{2}{\alpha} \leq 1 + \frac{4}{n} = 1 + o(1)$. □

5.4.2 Bounding the Price of Anarchy

We aim for an upper bound on the price of anarchy. The following observation is the key to show that a Nash equilibrium does not have many more edges than a tree.

5.16 Proposition. *A Nash equilibrium is chord-free.*

Proof. Selling a chord $e = \{v, w\}$ from a cycle $C = (v, \ldots, w, \ldots, v)$ does not increase the relevance of any edge for any player. For, let C' be a cycle which contains e. Then $C' - e$ also forms a cycle with a part of C, say $(v, \ldots, w)$. Hence, if the graph is bridgeless, removing a chord would decrease the player's building cost without increasing the disconnection cost. Let now the graph contain a bridge e'. Due to the decrease in the denominator of the disconnection cost, removing a chord impairs the disconnection cost. However, the player owning the chord, say v, would rather remove the chord and instead build an edge to form a new cycle containing e'. The only case where this is impossible is when v is one endpoint of the bridge $e' = \{v, u\}$, and u is a leaf vertex. Then, a double-edge between v and u would be needed, which is not allowed unless we use a multigraph.

We consider this case now and show that we in fact do not need a multigraph. By selling the chord, the disconnection cost for v increases

by $\frac{1}{m\,(m-1)}R(v)$. If this increase is strictly smaller than α, we are done. Hence assume $\frac{1}{m\,(m-1)}R(v) \geq \alpha$ now. Edge $\{v,u\}$ has relevance $n-1$ for u. The positions of v and u imply $R(u) = R(v) + (n-1) - 1$. If u builds an edge to any other vertex, save v, edge e' is put on a cycle. The improvement in disconnection cost for u due to such an edge is at least

$$
\begin{aligned}
&\frac{1}{m}R(u) - \frac{1}{m+1}(R(u) - (n-1)) \\
&= \frac{1}{m}(R(v) + n - 2) - \frac{1}{m+1}(R(v) - 1) \\
&= \left(\frac{1}{m} - \frac{1}{m+1}\right) R(v) + \frac{n-2}{m} + \frac{1}{m+1} \\
&= \frac{1}{m\,(m+1)}R(v) + \frac{n-2}{m} + \frac{1}{m+1} \\
&= \left(\frac{1}{m\,(m-1)} + \frac{1}{m\,(m+1)} - \frac{1}{m\,(m-1)}\right) R(v) + \frac{n-2}{m} + \frac{1}{m+1} \\
&\geq \alpha - \frac{1}{m}\left(\frac{1}{m-1} - \frac{1}{m+1}\right) R(v) + \frac{n-2}{m} + \frac{1}{m+1} \\
&\geq \alpha - \frac{1}{m}\left(\frac{1}{m-1} - \frac{1}{m+1}\right) \frac{n\,(n-1)}{2} + \frac{n-2}{m} + \frac{1}{m+1} \\
&\geq \alpha - \left(\frac{1}{m-1} - \frac{1}{m+1}\right) \frac{n-1}{2} + \frac{n-2}{m} + \frac{1}{m+1} \\
&= \alpha - \frac{2}{(m-1)\,(m+1)} \frac{n-1}{2} + \frac{n-2}{m} + \frac{1}{m+1} \\
&\geq \alpha - \frac{1}{m+1} + \frac{n-2}{m} + \frac{1}{m+1} > \alpha \ .
\end{aligned}
$$

So u has an incentive to buy an additional edge, a contradiction to Nash equilibrium. □

The next two are graph-theoretic results. The first is a straightforward adaption of a result (and its proof) on vertex-connectivity to edge-connectivity; see, e.g., [40, Prop. 3.1.3] for the version for vertex-connectivity.

5.17 Proposition. *Any bridgeless connected graph can be constructed from a cycle by successively adding paths of the form* $(u, e_1, v_1, \ldots, v_k, e_{k+1}, w)$,

where u, w are vertices of the already constructed graph and $v_1, \dots, v_k$ are zero or more new vertices.

Proof. Clearly, any graph that was constructed in this manner is connected and bridgeless. Now let G be connected and bridgeless and H a subgraph of G that is constructible in this manner, chosen such that it has a maximum number of edges among all such subgraphs. Since G contains a cycle, H is not empty. Also, H is an induced subgraph since $H + e$ is also constructible for any edge e. If $H \neq G$, then since G is connected, there is an edge $e = \{v, w\}$ with $v \notin V(H)$ and $w \in V(H)$. Since G is bridgeless, this edge is on a cycle $C = (w, e, v = v_1, \dots, v_k, w)$. Let v_i be the first vertex with $v_i \in V(H)$. Then $P := (w, \dots, v_i)$ is a path of the form used in the construction, and so $H + P$ is constructible and has more edges than H, a contradiction. □

5.18 Proposition. *A chord-free graph on n vertices contains no more than $3n = O(n)$ edges.*

Proof. Let G be a chord-free graph, w.l.o.g. being connected. We first consider the case that G is bridgeless. By the previous proposition, G can be constructed from a cycle on, say, N_0 vertices, by successively adding paths of the form $(u, e_1, v_1, \dots, v_k, e_{k+1}, w)$, where u, w are vertices of the already constructed graph and $v_1, \dots, v_k$, $k \in \mathbb{N}_0$, are zero or more new vertices. For any two vertices u, w in the already constructed graph, there is a cycle C with $u, w \in V(C)$. Since G is chord-free, we may not add a path (u, e_1, w). Hence $k \geq 1$ in each step, i.e., at least one new vertex is added. It follows that there are at most $t \leq n - N_0 =: N_1$ steps in this construction. Let n_i and m_i be the number of new vertices and edges, respectively, inserted in step i. Then $m_i = n_i + 1$ for each $i \in [t]$ and so we add $\sum_{i=1}^{t} m_i = \sum_{i=1}^{t} (n_i + 1) = N_1 + t \leq 2N_1$ edges to the initial cycle. It follows that G has at most $N_0 + 2N_1 \leq 2N_0 + 2N_1 = 2n$ edges.

If G is not bridgeless, we consider each of its BCCs; these correspond to vertices of the bridge tree. Altogether, they cannot contribute more than $2n$ edges. In addition, there are at most $n - 1$ edges, namely bridges of G. So we have a bound of $2n + n - 1 \leq 3n$. □

5.19 Corollary. *A Nash equilibrium has at most $3n = O(n)$ edges.*

Proof. Follows from Prop. 5.16 and 5.18. □

Now we know that the total building cost in a Nash equilibrium is $O(n\alpha)$, and so of the same order as the optimal social cost. In order to bound the price of anarchy, we are left with bounding the disconnection cost. To this end, we make use of the bridge tree. The following is a corollary to Lem. 5.3.

5.20 Corollary. *The disconnection cost is bounded by* $n \operatorname{diam}(\widetilde{G})$.

Proof. We have by Lem. 5.3:

$$\begin{aligned}&\frac{1}{m}\sum_{v\in V}\sum_{e\in E}\operatorname{rel}(e,v) = \frac{1}{m}\sum_{v\in V}R(v) \le \frac{1}{m}\sum_{v\in V}(n-1)\operatorname{diam}(\widetilde{G})\\ =&\frac{n}{m}(n-1)\operatorname{diam}(\widetilde{G}) \le n\operatorname{diam}(\widetilde{G})\ .\end{aligned}$$ □

5.21 Lemma. *The bridge tree of a Nash equilibrium has diameter* $O(\alpha)$.

Proof. Let G be a Nash equilibrium. Let $\widetilde{P} = (v_0, e_1, v_1, \ldots, e_\ell, v_\ell)$ be a path in the bridge tree $\widetilde{G}$ connecting two leaves v_0 and v_ℓ. Let $\bar{\ell} := \lceil \frac{\ell}{2} \rceil \ge 1$. Then at least one of the following is true (recall the convention on p. 137 regarding vertex-counting in the bridge tree):

– At least $\lceil \frac{n}{2} \rceil$ vertices lie beyond $e_{\bar{\ell}}$ from the view of v_0.

– At least $\lceil \frac{n}{2} \rceil$ vertices lie beyond $e_{\bar{\ell}}$ from the view of v_ℓ.

Let us assume the first; the other case can be treated alike. Let $v := v_0$ and $w := v_\ell$ and recall that we may treat vertices of the bridge tree $\widetilde{G}$ as single players with respect to building of new links. Then $e_1, \ldots, e_{\bar{\ell}}$ for v have relevance at least $\lceil \frac{n}{2} \rceil$ each. So $\sum_{i=1}^{\bar{\ell}} \operatorname{rel}(e_i, v) \ge \bar{\ell}\,\frac{n}{2} \ge \frac{\ell}{2}\,\frac{n}{2} = \Omega(\ell n)$. By building $\{v, w\}$, player v would have a benefit in disconnection cost of at least $\frac{1}{m+1}\frac{\ell n}{4} \ge \frac{1}{3n+1}\frac{\ell n}{4} = \Omega(\ell)$, using the bound $m \le 3n$ from Cor. 5.19. Since the edge is not built, α is larger than this benefit, so $\ell = O(\alpha)$. □

5.22 Corollary. *The disconnection cost in a Nash equilibrium is* $O(n\alpha)$.

Proof. Follows from Cor. 5.20 and Lem. 5.21. □

5.23 Theorem. *The price of anarchy with a simple-minded adversary is bounded by* $O(1)$.

Proof. The building cost and the disconnection cost in a Nash equilibrium are both $O(n\alpha)$ by Cor. 5.19 and 5.22. The theorem follows with Prop. 5.5, which states that the optimum social cost is $\Theta(n\alpha)$. □

5.5 Smart Adversary

We remain in the unilateral case and consider an adversary that destroys an edge which separates a maximum number of vertex pairs. If there are several such edges, one of them is chosen uniformly at random. In other words, we replace the uniform probability distribution on the edges for one that is concentrated on the edges which cause maximum overall damage. Recall that $\mathrm{sep}(e)$ is the number of separated vertex pairs when edge e is deleted. Let $\mathrm{sep}_{\max} := \max_{e\in E} \mathrm{sep}(e)$ and $E_{\max} := \{e \in E;\ \mathrm{sep}(e) = \mathrm{sep}_{\max}\}$ and $m_{\max} := |E_{\max}|$. These are the edges of which each causes a maximum number of separated vertex pairs when it is deleted. We call them the *critical* edges. We have the individual and social cost, respectively:

$$C_v(\vec{G},\alpha) = \deg^{\mathrm{out}}(v)\,\alpha + \frac{1}{m_{\max}} \sum_{e\in E_{\max}} \mathrm{rel}(e,v) \qquad \text{for } v \in V,$$

$$C(G,\alpha) = m\,\alpha + \frac{1}{m_{\max}} \sum_{e\in E_{\max}} \mathrm{sep}_{\max} = m\,\alpha + \mathrm{sep}_{\max}\ .$$

If $\mathrm{sep}_{\max} = 0$, then the graph is bridgeless and all edges are critical – however, their removal does not separate any vertex pairs. If $\mathrm{sep}_{\max} > 0$, then there are one or more critical edges, and each of them is a bridge. If there are multiple critical edges, then for any two of them, say e, e', we have $\nu(e) = \nu(e')$. Recall that if e is a bridge, $\nu(e)$ denotes the number of vertices in the smaller component of $G - e$, or $\frac{n}{2}$ if both components are of equal size. Recall also the convention on vertex-counting in the bridge tree on p. 137.

5.24 Proposition. *If there are more than one critical edges, they form a subgraph that is a star in the bridge tree $\widetilde{G}$.*

Proof. Let $\mathrm{sep}_{\max} > 0$. For any two distinct bridges e and e', one component of $G - e$ is strictly contained in one component of $G - e'$.

Therefore, with multiple critical edges, $\nu(e) < \frac{n}{2}$ for all $e \in E_{\max}$, and so also for all other bridges (since they have smaller $\nu(\cdot)$ value). In other words, there is always a small and a large component of $G - e$, with e being a bridge.

Let $P = (v_0, e_1, v_1, \ldots, v_{\ell-1}, e_\ell, v_\ell)$ be a path in the bridge tree $\widetilde{G}$ with e_1 and e_ℓ being distinct critical edges. First assume that v_ℓ is in the larger component of $G - e_\ell$. Then v_0 is in the smaller component of $G - e_1$. Then the smaller component of $G - e_2$ cannot contain v_0, since otherwise $\nu(e_1) < \nu(e_2)$, and e_1 would not be critical. So the component of $G - e_2$ containing v_0 is the larger one, and then the same holds for the component of $G - e_\ell$ containing v_0. This contradicts that v_ℓ is in the larger component of $G - e_\ell$. We can carry out the same argument with v_0 and e_1. Summarizing, now we know that the smaller component of $G - e_1$ is located 'before' P and that the smaller component of $G - e_\ell$ is located 'beyond' P.

If $\ell \geq 3$, then there is an edge f between e_1 and e_ℓ on P. The smaller component of $G - f$ strictly contains either the smaller component of $G - e_1$ or $G - e_\ell$. Since $\nu(e_1) = \nu(e_2)$, we have thus in particular, $\nu(f) > \nu(e_1)$, a contradiction that e_1 is critical. Hence there is no such edge f, and so $\ell = 2$. Since this holds for all pairs (e_1, e_ℓ) of critical edges, the set of all critical edges forms a star (in the bridge tree). □

5.25 Proposition. *If there are $k \geq 2$ critical edges, say $E_{\max} = \{e_1, \ldots, e_k\}$, and e_1 is put on a cycle by an additional edge, but not $e_2, \ldots, e_k$, then the new critical edges are $e_2, \ldots, e_k$. If $k \geq 3$ and the additional edge puts e_1 and e_2 on a cycle, but not $e_3, \ldots, e_k$, then the new critical edges are $e_3, \ldots, e_k$.*

Proof. An additional edge e only changes the $\nu(\cdot)$ value of those edges which are put on a cycle by e, namely it reduces them to 0. Hence, none of the edges in $\{e_2, \ldots, e_k\}$ (or $\{e_3, \ldots, e_k\}$) becomes less attractive for the adversary when e is added. Also no other edge becomes more attractive by the addition of e, since no $\nu(\cdot)$ value increases. □

5.5.1 Optima, Equilibria, and the Price of Stability

5.26 Proposition. *Optima are exactly as for the simple-minded adversary, namely cycle or star, for the same ranges of α. The optimum social cost is $\Theta(n\alpha)$.*

Proof. With the same argument as for the simple-minded adversary, an optimum is either the cycle or a tree. The cycle has social cost $n\alpha$. Since $\mathrm{sep}(e) = 2\nu(e)\,(n - \nu(e))$ for each $e \in E$, a tree with minimum social cost is again a star, with social cost $(n-1)\,\alpha + 2 \cdot 1 \cdot (n-1) = (n-1)\,(\alpha + 2)$. The proposition follows as in the proof of Prop. 5.5. □

We can as before construct simple-structured Nash equilibria for all ranges of α, similar to Prop. 5.12.

5.27 Proposition.

(i) If $\alpha \leq \frac{n}{2}$, then the cycle with all edges pointing in the same direction (either all clockwise or all counter-clockwise) is a Nash equilibrium.

(ii) If $\alpha \geq 1$, then a star with all edges pointing outwards is a Nash equilibrium.

(iii) If $\alpha \geq \frac{n}{2}$, then a path with all edges pointing to the nearest end (in case of even n, the middle edge having arbitrary orientation) is a Nash equilibrium with social cost $\Theta(n\alpha + n^2)$.

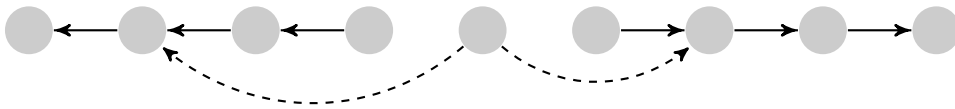
Proof. (i) There is no incentive to build additional edges, since the cycle already has 0 disconnection cost. Each vertex can sell only one edge, so that the graph will be a path. This creates a disconnection cost for the seller of $\frac{n}{2}$ if n is even, and $\frac{1}{2}\left(\lfloor \frac{n}{2} \rfloor + \lceil \frac{n}{2} \rceil\right) = \frac{n}{2}$ if n is odd; in both cases it is at least α. Buying further edges increases building cost to at least the previous value (which was α) but cannot yield a better disconnection cost than the cycle does.

(ii) Since all edges point outwards, we only have to show that building additional edges is not beneficial for the builder. We include the possibility of multiple edges, i.e., a multigraph, in the analysis. If an outer vertex connects to another outer vertex, her improvement in disconnection cost is

$$\frac{(n-1) + (n-2) \cdot 1}{n-1} - \frac{n-3}{n-3} = 1 - \frac{1}{n-1} \leq \alpha \ .$$

The same holds if the outer vertex builds another connection to the center vertex. The center vertex building an additional edge (to an outer vertex) experiences improvement 0 in disconnection cost. Further

a. A path with all edges pointing to the nearest end.



b. Center vertex cannot improve by exchanging her edges.

Figure 5.2. Nash equilibrium if $\alpha \geq \frac{n}{2}$. Critical edges are drawn dashed. Disconnection cost for the center vertex is $\lfloor \frac{n}{2} \rfloor$ in both cases, which is 4 here.

edges bring no benefit, no matter who the builder is, save the last one, which can bring an improvement of $1 \leq \alpha$.

(iii) The social cost of the path is $(n-1)\,\alpha + \lfloor \frac{n}{2} \rfloor \lceil \frac{n}{2} \rceil = \Theta(n\alpha + n^2)$. The adversary removes the one or two – depending on whether n is even or odd – middle edges. The disconnection cost for each player is $\frac{n}{2} \leq \alpha$ if n is even and at most $\frac{1}{2}\left(\lfloor \frac{n}{2} \rfloor + \lceil \frac{n}{2} \rceil\right) = \frac{n}{2} \leq \alpha$ if n is odd. Hence, there is no incentive for any player to build more edges than she currently owns, even after exchanging the currently built edges for others.

Now consider that a player v sells one (or two) of her edges and buys one (or two) different ones instead. First consider that one edge is exchanged. Since all edges point outwards, the part of the path that becomes disconnected from v does not contain the critical edge(s). So, after reconnecting it with v via a new edge, there are as many vertices on both sides of the formerly critical edge(s) as before the exchange. No separation value increases. Hence the formerly critical edges remain critical. They also maintain their relevance for v. With the same argument, if the exchanged edge itself was critical, the new one will be critical as well, also with the same relevance for v.

When two edges are exchanged, v is the center vertex, and in particular all critical edge(s) are among the exchanged ones, see Fig. 5.2. This again means that the disconnected parts do not contain critical edges, and so the exchange cannot change that each of the two edges

has $\lfloor \frac{n}{2} \rfloor$ vertices on the one and $\lceil \frac{n}{2} \rceil$ vertices on the other side, so they remain critical. Also, their relevance for v does not change. □

5.28 Remark. By Rem. 4.27, in the previous proposition, the cycle is a maximal Nash equilibrium, the star is a maximal Nash equilibrium for $\alpha > 1$, and the path is a maximal Nash equilibrium for $\alpha > \frac{n}{2}$.

5.29 Remark. The lower bound on the price of anarchy and the bound on the price of stability (Prop. 5.14 and Thm. 5.15, respectively) carry over.

Proof. The proofs for the simple-minded adversary work by comparing social cost of known Nash equilibria and optima. Optima are the same for the smart adversary. The proof of Prop. 5.14 uses that a star is a Nash equilibrium for large n. We have the same prerequisite for the smart adversary.

The proof of Thm. 5.15 uses that the cycle is a Nash equilibrium for $\alpha \leq \frac{n}{2}$ and that a star is a Nash equilibrium for $\alpha \geq \frac{n}{2}$. Both is also given for the smart adversary. □

5.5.2 Bounding the Price of Anarchy

The proof of the following is even easier than previously:

5.30 Proposition. *A Nash equilibrium is chord-free.*

Proof. Removing a chord does not change the relevance of any edge, nor does it change $\text{sep}_{\max}$, hence it does not change $E_{\max}$. Selling a chord so is always beneficial. □

With Prop. 5.18, it follows immediately:

5.31 Corollary. *A Nash equilibrium has only $O(n)$ edges.* □

We are again left with bounding the disconnection cost of Nash equilibria. This requires some effort and is accomplished in the following proof.

5.32 Theorem. *The price of anarchy with a smart adversary is $O(1)$.*

Proof. By Prop. 5.1, we only have to consider $\alpha \leq n$. Fix a Nash equilibrium with $\mathrm{sep}_{\max} > 0$. The aim is to show $\mathrm{sep}_{\max} = O(n\alpha)$. Let first be $m_{\max} \geq 3$. Fix two critical edges e_1 and e_2, and set $n_0 := \nu(e_1)$. For each $i \in \{1,2\}$ fix a player v_i in the smaller component of $G - e_i$. Then for each $i \in \{1,2\}$ we have $\mathrm{rel}(e_i, v_i) = n - n_0$ and $\mathrm{rel}(e, v_i) = n_0$ for all critical edges $e \neq e_i$; recall that all critical edges have the same $\nu(\cdot)$ value. Building $\{v_1, v_2\}$ puts e_1 and e_2 on a cycle and leaves the other $m_{\max} - 2$ critical edges critical by Prop. 5.25. For each $i \in \{1,2\}$,[4] player v_i has her disconnection cost decreased by:

$$\begin{aligned}
&\frac{1}{m_{\max}} \sum_{e \in E_{\max}} \mathrm{rel}(e, v_i) - \frac{1}{m_{\max} - 2} \sum_{\substack{e \in E_{\max} \\ e \notin \{e_1, e_2\}}} \mathrm{rel}(e, v_i) \\
&= \frac{1}{m_{\max}} \left((m_{\max} - 1)\, n_0 + n - n_0 \right) - \frac{1}{m_{\max} - 2} (m_{\max} - 2)\, n_0 \\
&= \frac{1}{m_{\max}} \left((m_{\max} - 2)\, n_0 + n \right) - n_0 = \frac{1}{m_{\max}} (n - 2n_0) \ .
\end{aligned}$$

Since we are in a Nash equilibrium, this is at most α. Since $n \geq m_{\max} n_0$, we have $n - 2n_0 \geq (m_{\max} - 2)\, n_0$, and so it follows $\alpha \geq (1 - \frac{2}{m_{\max}})\, n_0 \geq \frac{1}{3} n_0$. Moreover, it follows $m_{\max}\alpha + n_0 \geq n - n_0$. With these two inequalities at hand, we can bound $\mathrm{sep}_{\max}$. We have

$$\begin{aligned}
\mathrm{sep}_{\max} &= 2n_0\,(n - n_0) \leq 2n_0\,(m_{\max}\alpha + n_0) \leq 2\,(n_0 m_{\max}\alpha + 9\alpha^2) \\
&\leq 2\,(n\alpha + 9\alpha^2) = O(n\alpha + \alpha^2) = O(n\alpha) \ .
\end{aligned}$$

Now we consider the case $m_{\max} = 2$. A player can make the two critical edges part of a cycle by building an additional edge. The difficulty lies in that new critical edges, with a smaller separation value, can emerge. We will have to put some more effort into estimating the improvement in disconnection cost that a player is able to achieve by building another edge. Consider the bridge tree. There are two subtrees T_1 and T_2 that are connected to the rest by the two critical edges e_1 and

[4] It would suffice to restrict to $i = 1$ or $i = 2$. However, we point out all arguments that are symmetric in the sense that both endpoints would like to build the edge. This is also the reason why we treat the cases $m_{\max} = 2$ and $m_{\max} = 1$ separately. This is interesting for the bilateral case discussed in Sec. 5.6.

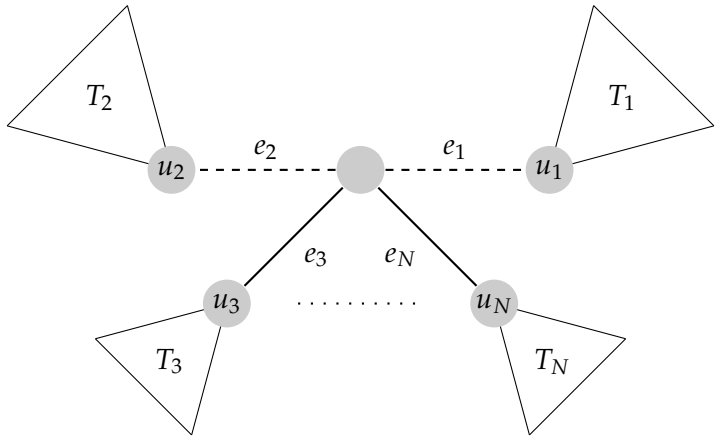


Figure 5.3. Schematic view of the bridge tree with two critical edges e_1 and e_2, drawn dashed. Subtrees are represented by triangles.

e_2, respectively. They both have $n_0 := \nu(e_1) = \nu(e_2)$ vertices. There may be more subtrees $T_3, \dots, T_N$ connected by $e_3, \dots, e_N$ to the center vertex. To streamline notation, we often write T_k instead of $V(T_k)$, $k \in [N]$, when we refer to the set of vertices of a tree. Fig. 5.3 depicts the situation. Fig. 5.4 on the facing page shows how a new edge would put e_1 and e_2 on a cycle.

First assume that we can arrange $v_1 \in T_1$ and $v_2 \in T_2$ such that after building $\{v_1, v_2\}$, there are no critical edges in T_1 nor in T_2. If there are no subtrees except T_1 and T_2, i.e., if $N = 2$, this means that we can make the graph bridgeless by the additional edge. The improvement in disconnection cost for v_1 (and also for v_2) of building $\{v_1, v_2\}$ is hence their original disconnection cost, i.e., $\frac{1}{2}(n - n_0 + n_0) = \frac{1}{2}n = \Omega(n)$, and so $\alpha = \Omega(n)$ and by Prop. 5.1 we are done. If $N \geq 3$, then critical edges emerge in one or more of the $T_3 + e_3, \dots, T_N + e_N$ after building. Fix $k \in \{3, \dots, N\}$. Since e_k is not critical without the new edge, we have $|T_k| < n_0$ or $|T_k| \geq \lceil \frac{n}{2} \rceil$. The latter can be excluded, since it would imply that the smaller component of $G - e_k$ includes T_1 and the center vertex, and so $\nu(e_k) > n_0 = \nu(e_1)$, in which case e_1 would not be critical. Moreover, we have $|T_k| \leq n - 2n_0 < n - 2|T_k|$, so $|T_k| < \frac{1}{3}n$. For a player in T_1 (or T_2), a critical edge in $T_k + e_k$ can have relevance at most $|T_k|$ and so no more than $\frac{1}{3}n$. The improvement in disconnection cost for v_1 (and also for v_2) gained by building $\{v_1, v_2\}$ is hence at least the original disconnection cost minus $\frac{1}{3}n$, i.e., $\frac{1}{2}(n - n_0 + n_0) - \frac{1}{3}n = \Omega(n)$, and so $\alpha = \Omega(n)$, and by Prop. 5.1 we are done.

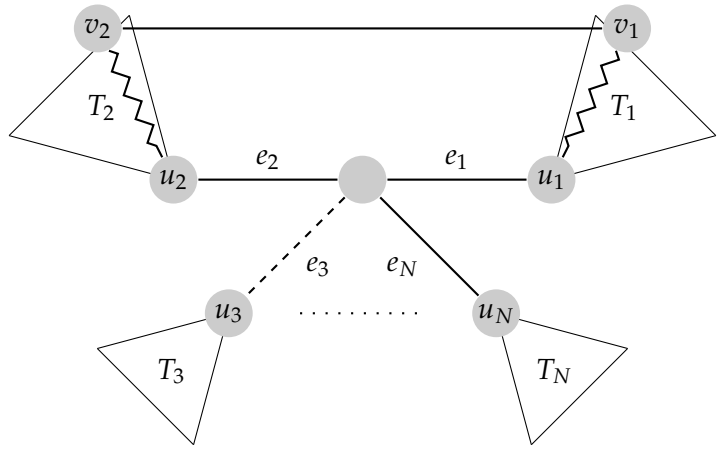


Figure 5.4. How e_1 and e_2 are put on a cycle by a new edge $\{v_1, v_2\}$. Paths that are part of the new cycle and located inside T_1 and T_2 are depicted as zig-zag paths. New critical edges can emerge, e.g., e_3 can become critical.

Now consider that for all choices of $v_1 \in T_1$ and $v_2 \in T_2$, building $\{v_1, v_2\}$ induces a critical edge in at least one of T_1 or T_2. For each $i \in \{1, 2\}$ we can do the following. Let u_i be the vertex where T_i is connected to the rest of the graph and consider T_i being rooted at u_i. Let P_i be a path starting at u_i and ending at one of the leaves of T_i, say w_i, such that the path always descends into a subtree that has a maximum number of vertices, as shown in Fig. 5.5 on the next page. If we choose $v_i := w_i$, $i = 1, 2$, then each P_i does not contain a critical edge when we build $\{v_1, v_2\}$, since these paths then both are located on a cycle. However, by assumption, there is a critical edge f in, say T_1. By construction of P_1, we have $\nu(f) \leq \frac{n_0}{2}$. So, player v_1 (and also v_2) can reduce her disconnection cost to no more than $\frac{n_0}{2}$. It follows that the improvement in disconnection cost is at least $\frac{1}{2}(n - n_0 + n_0) - \frac{1}{2}n_0 = \frac{1}{2}(n - n_0)$, which is at most α, since we are in a Nash equilibrium. It follows $\mathrm{sep}_{\max} = 2n_0\,(n - n_0) \leq 2n_0 \cdot 2\alpha = O(n\alpha)$.

The remaining case of $m_{\max} = 1$ can be treated similarly. Let e_1 be the critical edge and T_1 the subtree with $n_0 := \nu(e_1)$ vertices. There are zero or more additional subtrees, say $T_2, \ldots, T_N$. If there are zero such trees, define $T_2 := \widetilde{G} - T_1$, which consists of just one vertex in the bridge tree then (but can consist of multiple vertices in G). Let the ordering be such that $|T_2| \geq |T_k|$ for all $k \in \{3, \ldots, N\}$. Then we argue similar to before with T_1 and T_2 in the roles of the former subtrees of the same name. Assume first that we can find $v_1 \in T_1$ and $v_2 \in T_2$ such that building $\{v_1, v_2\}$ does not induce any critical edges

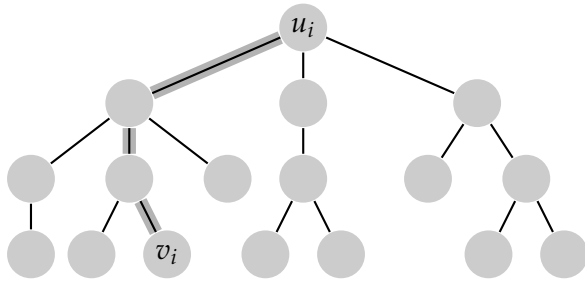


Figure 5.5. Detailed view of T_i for one $i \in \{1,2\}$. Path P_i is highlighted. Recall that vertices of the bridge tree are counted according to the size of the respective BCCs. Here, in this example, we assume that each vertex counts 1. The path is drawn accordingly, i.e., always descending into a subtree with a maximum number of vertices.

in T_1 nor T_2. If $N \leq 2$, then we can make the graph bridgeless and this means an improvement for v_1 of at least $n - n_0$, and so $\text{sep}_{\max} = 2n_0(n - n_0) \leq 2n_0\alpha = O(n\alpha)$. If $N \geq 3$, then fix $k \in \{3, \ldots, N\}$. We have $|T_k| < n_0$. Moreover, we have $|T_k| \leq n - (|T_2| + n_0) \leq n - 2|T_k|$, and so $|T_k| \leq \frac{1}{3}n$. Then building $\{v_1, v_2\}$ reduces the disconnection cost of v_1 to no more than $\frac{1}{3}n$. This means an improvement for v_1 of at least $n - n_0 - \frac{1}{3}n \geq \frac{2}{3}n - \frac{1}{2}n = \Omega(n)$, and we are done by Prop. 5.1.

If each choice of v_1 and v_2 induces a critical edge in T_1 or T_2, we can, as before, show that by a careful choice of these vertices, building $\{v_1, v_2\}$ reduces the disconnection cost for v_1 (and v_2) to at most $\frac{n_0}{2}$. Player v_1 originally has disconnection cost $n - n_0 \geq n_0$, so she experiences an improvement of at least $\frac{n_0}{2}$. (Player v_2 originally has disconnection cost n_0, so she as well experiences an improvement of at least $\frac{n_0}{2}$.) It follows $n_0 \leq 2\alpha$ and so $\text{sep}_{\max} = 2n_0(n - n_0) \leq 4\alpha n = O(n\alpha)$. □

5.6 Bilateral Case

Now we study the adversary model in the bilateral setting as per Def. 4.1(ii). We proceed in roughly the same order as in the sections before: first we briefly consider optima and equilibria, then we bound the price of anarchy. Our equilibrium concept is pairwise stability. In Sec. 5.6.2 we show that this is equivalent to pairwise Nash equilibrium for the simple-minded adversary. We do not succeed in proving a similar result for the smart adversary – however, we give a tight lower bound (for $\alpha > 2$) on the price of anarchy in Sec. 5.6.1, which also holds for pairwise Nash equilibrium.

Prop. 5.1 holds independently of the equilibrium concept. The cycle is optimal for $\alpha \leq n - 1$ and a star is optimal for $\alpha \geq n - 1$ with social

cost $\Theta(n\alpha)$. Ranges for α are different from those in the unilateral case, accounting for the factor 2 in building cost, cf. p. 131. Otherwise, arguments are the same. Our existence results for maximal Nash equilibria, Rem. 5.13 and 5.28, also carry over, by the transformation rules in Sec. 4.4.3.

5.6.1 Bounding the Price of Anarchy

Simple-Minded Adversary

Recall that we have shown in Prop. 5.16 that a Nash equilibrium is chord-free. The proof does not fully carry over to the bilateral case, since it contains an argument of the form "then the player would rather build a different link instead." Yet, we can use the idea of that proof to show chord-freeness if α is not too small. For small α, we can show a bound on the number of edges by a different simple argument.

5.33 Proposition. *Let a pairwise stable graph G be given.*

(i) If $\alpha > \frac{1}{2}$, then G is chord-free and hence only has $O(n)$ edges.

(ii) In general, G is chord-free (with $O(n)$ edges) or has at most $\frac{n}{\sqrt{2\alpha}} + 1$ edges.

Proof. If G is bridgeless, selling a chord is beneficial since disconnection cost 0 is maintained. So for both parts we assume that G contains bridges.

(i) The impairment in disconnection cost for a player v of selling a chord is only due to the change in the denominator of the disconnection cost and is precisely $\frac{1}{m\,(m-1)}\,R(v)$, which is upper-bounded by $\frac{1}{2}$ since $R(v) \leq \frac{n\,(n-1)}{2}$. Hence if α is larger than that, there is an incentive to sell the chord.

(ii) Let G possess a chord. This means that any of its two endpoints, say v, deems it being no impairment to pay α for this edge, hence $\frac{1}{m\,(m-1)}\,R(v) \geq \alpha$. It follows

$$\frac{n^2}{2} \geq \frac{n\,(n-1)}{2} \geq R(v) \geq m\,(m-1)\,\alpha \geq (m-1)^2\,\alpha \ ,$$

hence $\frac{n}{\sqrt{2\alpha}} + 1 \geq m$. □

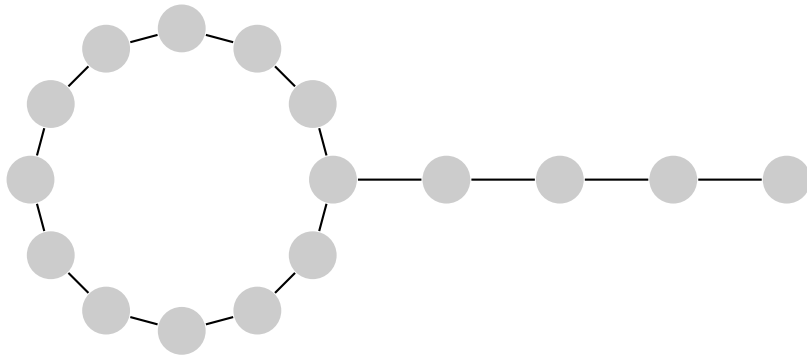


Figure 5.6. Cycle with path attached, here $n = 16$ and $\ell = 4$.

As for bounding disconnection cost, Lem. 5.21 is no longer true, but Lem. 5.3 is. We can at least show a bound of $O(1 + \sqrt{n/\alpha})$ on the price of anarchy; we do not know whether it is tight.

5.34 Lemma.

(i) The diameter of the bridge tree of a pairwise stable graph is $O(\sqrt{n\alpha})$.

(ii) For $\alpha = 1$, *the bridge tree can have diameter* $\Omega(\sqrt{n})$, *even if the graph is pairwise stable.*

Proof. (i) Building an edge that puts a path in the bridge tree of length $\ell \geq 1$ on a cycle brings to *both endpoints* at least the ΔR of that path alone, i.e., $\sum_{k=1}^{\ell} k = \frac{\ell(\ell+1)}{2}$. So the edge brings an improvement in disconnection cost of at least $\frac{1}{m+1}\Delta R \geq \frac{1}{3n+1}\frac{\ell(\ell+1)}{2}$, hence $7n\alpha \geq \ell(\ell+1) \geq \ell^2$. (We use the $3n$ bound from Prop. 5.18 here.)

(ii) Consider a cycle with a path of length ℓ attached to it with one of its ends, as shown in Fig. 5.6. Let n be the total number of vertices and let $\frac{1}{n}\frac{\ell(\ell+1)}{2} \leq \alpha \leq \frac{1}{n}\frac{((n-\ell)-1)(n-\ell)}{2}$; such an α exists if $n \geq 3\ell$. The bridge tree has diameter ℓ. Because of the lower bound on α, no vertex on the cycle wishes to connect to a vertex on the path, and also no vertex on the path wishes to connect to a vertex that is located away from the cycle. Because of the upper bound on α, it can also be shown easily that no two neighboring vertices on the cycle wish to sell the edge between them. Trivially, no edge on the path will be sold. Hence this

graph is pairwise stable. However, we can choose $n \geq 9$, $\ell := \lfloor \sqrt{n} \rfloor$, and $\alpha := 1$, and so have a diameter of $\Omega(\sqrt{n})$. □

5.35 Remark. The example in the proof of Lem. 5.34(ii) does not prove a price of anarchy beyond $\Theta(1)$.

Proof. The total disconnection cost is

$$\frac{1}{n} 2 \sum_{k=1}^{\ell} (n-k)\, k = \frac{\ell\,(\ell+1)}{n} \left(n - \frac{2\ell+1}{3} \right) \ .$$

For a lower bound on the price of anarchy, we have to divide this by $\Theta(n\alpha)$, so we choose α as small as possible, i.e., $\alpha = \frac{1}{n} \frac{\ell\,(\ell+1)}{2}$. Then we divide the disconnection cost by $\Theta(n\alpha) = \Theta(\ell\,(\ell+1))$ and receive a lower bound on the price of anarchy of only

$$\Theta\left(\frac{1}{n} \left(n - \frac{2\ell+1}{3} \right) \right) = \Theta\left(1 - \frac{2\ell+1}{3n} \right) = O(1) \ . \qquad \square$$

5.36 Theorem. *The price of anarchy for a simple-minded adversary in the bilateral case with pairwise stability is* $O(1 + \sqrt{n/\alpha})$.

Proof. By Prop. 5.33, building cost of a pairwise stable graph is $O(n\alpha)$ or $O((\frac{n}{\sqrt{2\alpha}} + 1)\,\alpha)$, both having a ratio of $O(1 + \frac{1}{\sqrt{\alpha}})$ to the optimum.

By Lem. 5.34(i) and Cor. 5.20, disconnection cost of a pairwise stable graph is $O(n\sqrt{n\alpha})$, having a ratio of $O(\sqrt{n/\alpha})$ to the optimum. □

Smart Adversary

The proof of Prop. 5.30, showing that a Nash equilibrium is chord-free, clearly carries over to the bilateral case and pairwise stability. This is easier compared to the simple-minded adversary, where we had to prove Prop. 5.33.

We always have the $O(n + \frac{n}{\alpha})$ bound of Prop. 5.1. Since $m = O(n)$, it improves to

$$O\left(1 + \frac{n}{\alpha}\right) \ . \tag{5.2}$$

The proof of Thm. 5.32 is almost completely symmetric in v_1 and v_2 and so at first appears to apply in the bilateral case as well, which

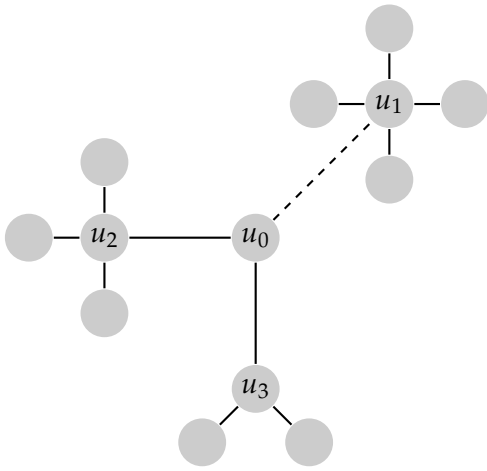


Figure 5.7. Three stars of sizes n_0, $n_0 - 1$, and $n_0 - 2$; here $n_0 = 5$. The n_0 players in the star around u_1 would like to put the one critical edge on a cycle, if $\alpha < n_0$. Building, e.g., $\{u_1, u_2\}$ would reduce their disconnection cost from $n - n_0$ to $n_0 - 2$, meaning an improvement of n_0. But no player from the stars around u_2 or u_3 is willing to cooperate.

would have meant an $O(1)$ bound. The case of exactly one critical edge, however, is not symmetric and provides an idea to a counterexample: if there is only one critical edge e_0, then it can happen that vertices in the smaller component of $G - e_0$ wish (desperately) to put e_0 on a cycle, but they cannot find a partner in the other component that is willing to cooperate. The following lower bound is tight by (5.2).

5.37 Theorem. *Consider the smart adversary and pairwise Nash equilibrium or pairwise stability. If $\alpha > 2$ and $n \geq 10$, then $\rho(n, \alpha) = \Omega(1 + \frac{n}{\alpha})$.*

Proof. First assume there is an integer $n_0 \geq 4$ such that $n = 3n_0 - 2$. Consider three stars S_i, $i = 1, 2, 3$ with center vertices u_i, $i = 1, 2, 3$, and n_0, $n_0 - 1$, and $n_0 - 2$ vertices, respectively. Connect the stars via an additional vertex u_0 and additional edges $\{u_0, u_i\}$, $i = 1, 2, 3$. This construction uses $3n_0 - 2$ vertices. See Fig. 5.7 for an illustration. Then $e_0 = \{u_0, u_1\}$ is the only critical edge and $n_0 = \Theta(n)$, namely slightly more than $\frac{1}{3}n$. We have a total disconnection cost of $2\nu(e_0)\,(n - \nu(e_0)) = 2n_0\,(n - n_0) = \Omega(n^2)$. We have a ratio to the optimum of $\Omega(1 + \frac{n}{\alpha})$. We are left to show that this graph is a pairwise Nash equilibrium, which implies pairwise stability. It is clear that no edge can be sold, since that would make the graph disconnected. Therefore we only have to ensure that no link can be added that would be beneficial for one endpoint and at least no impairment for the other one, i.e., we have to show (4.3).

An edge e that improves disconnection cost for some player has to put $\{u_0, u_1\}$ on a cycle. If e connects a vertex in S_1 with a vertex in S_2, then $\{u_0, u_3\}$ will become critical. For a vertex in S_2, this reduces disconnection cost from n_0 to $n_0 - 2$. So, since $\alpha > 2$, no vertex in S_2 agrees to build such an edge.

A similar situation holds if e connects a vertex in S_1 with a vertex in $S_3 + u_0$. It will result in $\{u_0, u_2\}$ becoming critical. For a vertex in $S_3 + u_0$, this reduces disconnection cost from n_0 to $n_0 - 1$. So, since $\alpha > 1$, no vertex in $S_3 + u_0$ agrees to build such an edge.

If $n + 2$ is not a multiple of 3, we can do a similar construction. We let n_0 be as large as possible so that $n \geq 3n_0 - 2$ and do the same construction as above. The remaining 1 or 2 vertices are connected directly to u_0. Then the previous arguments essentially carry over. □

If we consider $\alpha > 2$ a constant, then the previous theorem gives a lower bound of $\Omega(n)$. For $\alpha = \Omega(1)$ this is the worst that can happen for *any* adversary in this model. Hence the 'overall worst-case' is attained by the smart adversary in the bilateral case.

5.6.2 Convexity and Non-Convexity of Cost

Simple-Minded Adversary

We show that for the simple-minded adversary, individual cost functions are convex, hence pairwise Nash equilibrium and pairwise stability are equivalent. The following proposition is purely graph-theoretical.

5.38 Proposition. *Let $G = (V, E)$ be a connected graph, $v \in V$ a vertex, $e = \{v, w\} \in E$ an edge, and $F \subseteq E \setminus \{e\}$ a set of edges, each incident with v, so that $G' := G - F - e$ is still connected. Let B_1 be those edges that are non-bridges in G but bridges in $G - e$. Let B_2 be those edges that are non-bridges in $G - F$ but bridges in $G - F - e$. Then $B_1 \subseteq B_2$.*

Proof. Since G' is connected, there is a path $(v, e_1, v_1, \ldots, w)$ in G'. Then the cycle $C := (v, \ldots, w, e, v)$ is in $G - F$. By Prop. 5.6, we have $B_1 \subseteq E(C)$. Hence all edges in B_1 are on a cycle that is not destroyed by removal of F, so no edge in B_1 is made a bridge by removal of F. It follows $B_1 \subseteq B_2$. □

To streamline notation, we fix α and write C_v instead of $C_v(\cdot, \alpha)$ for the rest of this section.

5.39 Lemma. *The individual cost function C_v for the simple-minded adversary is convex for each $v \in V$.*

Proof. Let $v \in V$ and $w_1, \ldots, w_k \in V$. Let S be a strategy profile. We proceed by induction, as in the proof of Lem. 4.22. The case $k = 1$ is clear. Let $k > 1$ and set $S' := S - (v, w_1) - \ldots - (v, w_{k-1})$. We have to show that switching from S' to $S' - (v, w_k)$ increases cost for v at least as much as switching from S to $S - (v, w_k)$. Since $G(S')$ has fewer edges than $G(S)$, it suffices to consider changes in relevance $R(\cdot)$.

When removing $\{v, w_k\}$, relevance of zero or more edges changes from 0 to a positive value; these are precisely those edges which become bridges by the removal and which were no bridges before. No relevance is reduced by removal of edges.

Let B_1 be all those edges that become bridges by the switch from S to $S - (v, w_k)$, and let B_2 those that become bridges by the switch from S' to $S' - (v, w_k)$. Then $B_1 \subseteq B_2$ by Prop. 5.38. The increase in relevance from 0 to a positive value for $e \in B_1$ given S' is at least as high as when given S. In other words, while $\{v, w_1\}, \ldots, \{v, w_{k-1}\}$ are removed, the effect of all edges in B_1 becoming bridges is saved until the removal of $\{v, w_k\}$. □

Smart Adversary

The difficulty with the smart adversary is that its probability measure can change substantially when edges are removed. Indeed, exploiting this feature we show that the individual cost functions are *not* convex for the smart adversary.

5.40 Proposition. *Consider the graph in Fig. 5.8 on the facing page and the player v. Then C_v is not convex, since*

$$\begin{aligned} &C_v(G - e_1 - e_2) - C_v(G) \\ &< (C_v(G - e_1) - C_v(G)) + (C_v(G - e_2) - C_v(G)) \ . \end{aligned}$$

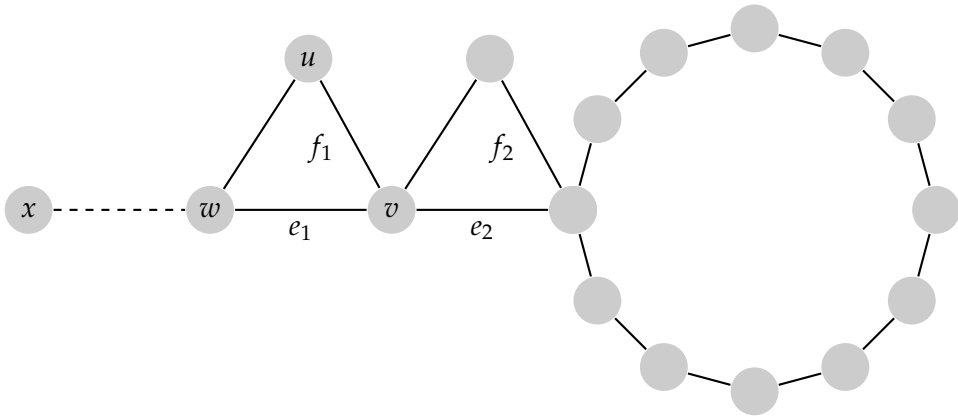


Figure 5.8. Example for a non-convex individual cost function C_v. The dashed edge is critical. Since the number of vertices on the cycle on the right is large enough, removal of e_i makes f_i critical for each $i \in \{1,2\}$.

Proof. Let k be the number of vertices on the cycle on the right. If k is large enough, removal of e_i makes f_i critical, for each $i \in \{1,2\}$. Removing both e_1 and e_2 makes f_2 critical. Thus we have:

$$\begin{aligned} C_v(G-e_1) - C_v(G) &= 3-1-\alpha = 2-\alpha \ , \\ C_v(G-e_2) - C_v(G) &= k-1-\alpha \ , \text{ and} \\ C_v(G-e_1-e_2) - C_v(G) &= k-1-2\alpha \ . \end{aligned}$$

So $C_v(G-e_1) - C_v(G) + C_v(G-e_2) - C_v(G) = 2+k-1-2\alpha = k+1-2\alpha$, which is strictly larger than $k-1-2\alpha$ by a difference of 2. □

This result is only partly satisfactory, since non-convexity is not shown on the set of pairwise stable strategy profiles. It remains unclear how to construct an example of a pairwise stable graph that is not a pairwise Nash equilibrium. In an attempt to make the example from Fig. 5.8 pairwise stable, we need $\alpha \leq 1$, or else u would sell $\{u,w\}$, which would make e_1 critical. However, then there is an incentive for x to put the critical edge on a cycle by building an additional edge, and no potential partner can decline such a request.

Yet, this example provides evidence that the smart adversary is in some respect a substantially different model than the sum-distance

model or the simple-minded adversary. For, these two have convex cost functions on the whole set of strategy profiles, which the smart adversary has not.

5.7 Summary

We considered distributed network formation in a model where players know that exactly one link will be deleted at some point in the future and each player v wishes to keep the number of other players to which v will become disconnected low. Link deletion happens according to a probability distribution, which is also known to the players, but which may depend on the network actually built. We call such a mapping from the set of networks to the set of probability distributions an adversary, since it models an entity which looks at the network and then makes a more or less random decision which link to delete. We considered two adversaries: a simple-minded adversary, who just picks one link at random and is modeled by the uniform distribution, and a smart adversary, who picks a link which separates a maximum number of vertex pairs. We proved an $O(1)$ bound on the price of anarchy for both adversaries in the unilateral case. Then we considered the bilateral case. We could only prove an $O(1 + \sqrt{n/\alpha})$ bound on the price of anarchy here, for the simple-minded adversary. We do not know whether it is tight. For the smart adversary, we could prove a tight lower bound of $\Omega(1 + \frac{n}{\alpha})$ for $\alpha > 2$. In other words, distributed bilateral network formation against a smart adversary can be really disadvantageous.

Conclusion and Future Directions

Network formation games are a vivid field of research. New interesting models and equilibrium concepts appear to come at the same rate as results concerning their analysis. We attempt an explanation why this is so:

- We have a framework that is rather simple on the one hand but on the other hand can be used to formulate a variety of models with quite different properties.
- The models offer clearance for many different equilibrium concepts.
- The topic has strong relations to graph theory.

We have seen how a new definition of indirect cost, in our case the disconnection cost, can provide new challenges and interesting phenomena. New definitions of indirect cost can be created in many different ways, for instance by combining existing definitions. We only point out three directions for future work here, which have a close connection to the model and results presented in this chapter:

1. Our proofs for the adversary model rely heavily on the restriction that the adversary only deletes one link. What happens if we allow two, three, or more links to be deleted?
2. The bilateral model shows a high price of anarchy (see Thm. 5.37), the highest possible for any adversary when $\alpha = \Omega(1)$. Will it be reduced when switching to a model with transfers, like the one introduced by Bloch and Jackson [14]?
3. An interesting indirect cost is maximum congestion. If each vertex v connects to each other w by a path P_{vw}, we can count, for each link e, how many of those paths traverse e. This is the congestion of e,

similar to the notion of congestion we know from Part I. What if we use the maximum congestion – for an optimal choice of paths $(P_{vw})_{v,w \in V}$ – as indirect cost? For a bridge e, the congestion is our familiar separation value $\mathrm{sep}(e)$. So we touched this topic already. However, for a non-bridge the congestion is not always zero, and so we did not cover that fully.

Appendix A

Basic Terminology and Notation

A.1 Miscellaneous

For an integer n, denote $[n] \coloneqq \{1, \ldots, n\}$. We usually write larger numbers in the text like this: 100,000,000. This example means 10^8, i.e., 100 million.

The transpose of a matrix A is denoted $A^\top$. Vectors in $\mathbb{R}^n$ are $(n \times 1)$-matrices. We write products between numbers, vectors, or matrices sometimes with a "$\cdot$", but usually without. For a set $U \subseteq \mathbb{R}^n$ denote $\overline{U}$ its closure.

Let $f : U \longrightarrow \mathbb{R}$ be a real-valued function on some set U and $V \subseteq U$ be a subset on which f is known to attain a maximum. By $\arg\max_{x \in V} f(x)$ we denote an arbitrary maximum point, i.e., an arbitrary $x \in V$ with the property that $f(x) \geq f(y)$ for all $y \in V$.

Regarding "O" and "Ω" notation, we use (for the results proved here) the following understanding. We write "$x = O(y)$" if there exists a constant $c > 0$ such that $x \leq cy$. The constant may only depend on other constants and is in particular independent of the non-constant quantities that constitute x and y. We do not implicitly require that some quantities have to be large. However, it may happen that the expression making up y is only defined when some quantities are large; the statement is then meant to hold only for those cases where the expression is defined. For example, in a statement of the form "Let $p \in \mathbb{N}_{\geq 1}$. Then we have ... and ... $= O(\frac{p}{\ln p})$" the "O" statement is only meant to hold for $p \in \mathbb{N}_{\geq 2}$.

Analogously to "O", we write "$x = \Omega(y)$" if there exists a constant $c > 0$ such that $x \geq cy$. Note that "O" indicates an upper bound, making no statement about a lower bound; while "Ω" indicates a lower bound, making no statement about an upper bound. We write $x = \Theta(y)$ if $x = O(y)$ and $x = \Omega(y)$; the constants used in the "O" and the "Ω" statement may be different, of course.

A.2 Graphs and Networks

A *graph* is a pair (V, E), where V is a finite set and $E \subseteq \binom{V}{2}$. A *directed graph* is a pair (V, E), where V is a finite set and $E \subseteq V \times V$. In both cases, we call elements of V *vertices* or *nodes*, and we call elements of E *edges* or *links* (we use both terms interchangeably). In case of a directed graph, we also use the terms *directed edge* or *directed link*. For a directed edge $e = (v, w)$ denote the *reverse edge* $e^{-1} := (w, v)$. Since E is a set, each edge can occur at most once. We sometimes call a graph an *undirected graph* to emphasize that it is not a directed graph. If not indicated otherwise, for a graph or directed graph on $n := |V|$ vertices, we assume $V = [n]$. The *adjacency matrix* of a graph $G = (V, E)$ on n vertices is an $n \times n$ matrix A over $\{0, 1\}$ such that $A_{vw} = 1$ if and only if $\{v, w\} \in E$. The adjacency matrix of a graph is symmetric. The *adjacency matrix* of a directed graph $G = (V, E)$ on n vertices is an $n \times n$ matrix A over $\{0, 1\}$ such that $A_{vw} = 1$ if and only if $(v, w) \in E$. It is not symmetric in general. For a directed graph $G = (V, E)$, define its *underlying undirected graph* $\widehat{G} = (V, \widehat{E})$ by $\widehat{E} := \{\{v, w\}\,;\, v, w \in V :\ (v, w) \in E \vee (w, v) \in E\}$.

Let $G = (V, E)$ be a graph or directed graph. We say that a vertex v and an edge e are *incident* if there exists a vertex w such that $e = \{v, w\}$ for a graph, or $e = (v, w)$ or $e = (w, v)$ for a directed graph. If e is an edge, we write $G - e$ for the graph or directed graph that results from G by removing e from E. We use the same notation for sets of edges. If v is a vertex, we write $G - v$ for the graph or directed graph that results from G be removing v from V and all incident edges from E. We use the same notation for sets of vertices.

For an undirected graph and a vertex v, the *degree* of v, denoted $\deg(v)$, is the number of edges that v is incident with, i.e., $\deg(v) :=$

$|\{\{v,w\}\,;\, w \in V\} \cap E|$. For a directed graph and vertex v, the *out-edges* of v are those edges that v is incident with and which point away from v, i.e., $\delta^{\text{out}}(v) \coloneqq (\{v\} \times V) \cap E$. The *in-edges* are defined analogously, namely $\delta^{\text{in}}(v) \coloneqq (V \times \{v\}) \cap E$. The *out-degree* of v is the number of edges that v is incident with and which point away from v, i.e., $\deg^{\text{out}}(v) \coloneqq |\delta^{\text{out}}(v)|$. The *in-degree* is defined analogously, namely $\deg^{\text{in}}(v) \coloneqq |\delta^{\text{in}}(v)|$.

Let $W = (v_0, e_1, v_1, \ldots, v_{\ell-1}, e_\ell, v_\ell)$ be a sequence consisting alternately of vertices and edges such that for all $i \in [\ell]$ we have $e_i = \{v_{i-1}, v_i\} \in E$ for a graph and $e_i = (v_{i-1}, v_i) \in E$ for a directed graph. Then W is called a *walk* of length ℓ. Sometimes, we omit specifying the edges, then a walk is denoted as a sequence of vertices. Denote $V(W) \coloneqq \{v_0, \ldots, v_\ell\}$ and $E(W) \coloneqq \{e_1, \ldots, e_\ell\}$. A *path* is a walk where all vertices are distinct. A *cycle* is a walk such that all edges are distinct, all vertices $v_0, \ldots, v_{\ell-1}$ are distinct, and $v_0 = v_\ell$. We say that two vertices v and w are *connected*, or we say that there exists a path between them, if there exists a path of the form $(v, \ldots, w)$. If $W = (v_0, \ldots, v_\ell)$ and $W' = (v'_0, \ldots, v'_\ell)$ are walks with $v_\ell = v'_0$, we denote the *concatenation* of W and W' by $W \circ W' \coloneqq (v_0, \ldots, v_\ell, \ldots, v'_\ell)$. For an undirected graph, $W^{-1} \coloneqq (v_\ell, e_\ell, \ldots, e_1, v_0)$ is the *reverse sequence* corresponding to W. The reverse sequence (in an undirected graph) is always a walk.

The *distance* between two vertices v and w, denoted $\text{dist}_G(v,w)$, is the length of a shortest path from v to w, or ∞ if there is no path between v and w. The *diameter* is the longest distance, i.e., $\text{diam}(G) \coloneqq \max_{v,w \in V} \text{dist}_G(v,w)$. We omit the "$G$" subscripts when it is obvious which graph is meant.

A graph or directed graph $H = (W, F)$ is called a *subgraph* of $G = (V, E)$ if $W \subseteq V$, $F \subseteq E$. If a subgraph is just given as "H", we write $V(H)$ for its vertices and $E(H)$ for its edges. A subgraph $H = (W, F)$ is called *induced* if $F = E \cap \binom{W}{2}$ for a graph and $F = E \cap (W \times W)$ for a directed graph. For a set of vertices $W \subseteq V$, there is exactly one induced subgraph on W, denoted $G[W]$. Sometimes, we identify W with $G[W]$ and call W a "subgraph".

We define the following notions concerning connectedness only for undirected graphs $G = (V, E)$. We call G *connected* if any two vertices $v, w \in V$ are connected, i.e., if there is a path in G between any two

vertices $v, w \in V$. We call it *disconnected* otherwise. A set $W \subseteq V$ is called *connected* if $G[W]$ is connected, and it is called *disconnected* otherwise. If $W \subseteq V$ is maximal under the condition of being connected, we call W a *connected component* or just *component*. A connected graph has exactly one connected component, namely G itself (or more precisely the set of all its vertices). An edge e is called a *bridge* if $G - e$ has more connected components than G; we know then that $G - e$ has exactly one more connected component than G. If G is connected and e is a bridge, we define $\nu(e)$ to be the number of vertices in the smaller connected component of $G - e$, or $\frac{|V|}{2}$ if both components have an equal number of vertices. If e is no bridge, we define $\nu(e) := 0$. We call a graph *bridgeless* if it does not contain any bridges. If $W \subseteq V$ is maximal under the condition that the induced subgraph $G[W]$ is connected and does not contain any bridges of $G[W]$, we call W a *bridgeless connected component*, abbreviated "BCC".

Sometimes, it is convenient to have multiple edges between a pair of vertices (even pointing in the same direction in the directed case). This is impossible for a graph or directed graph, since E is a set. To overcome this limitation, we introduce multigraphs and directed multigraphs. A *multigraph* is a pair (V, E), where V is a finite set and $E \subseteq \binom{V}{2} \times \mathbb{N}$. The elements of E are called *edges* or *links*. We call i the index of an edge $(\{v, w\}, i)$. We omit the index whenever it is irrelevant, and this will in fact be the case throughout this thesis. For instance, we write "$\{v, w\} \in E$" meaning that there exists $i \in \mathbb{N}$ such that $(\{v, w\}, i) \in E$. A *directed multigraph* is a pair (V, E) where V is a finite set and $E \subseteq V \times V \times \mathbb{N}$. The elements of E are called *edges* or *links*. We use the same simplified notation for directed multigraphs as for multigraphs, i.e., we omit the index whenever it is of no importance.

Most notions that we introduced for graphs and directed graphs carry over straightforwardly to multigraphs and directed multigraphs. In some places, we have to pay attention that two edges, although connecting the same pair of vertices, may be distinct on ground of their indices. In particular, a multigraph can have cycles of length 2, while this is impossible for graphs (it is possible for directed graphs, though).

We use "network" as an informal term referring to a graph, directed graph, multigraph, or directed multigraph possibly plus additional information, e.g., latency functions on the edges or certain vertices being marked as sources or terminals.

Appendix B

Experimental Results

We give plots and tables to display in detail our experimental findings.

- Hexagonal binning plots depicting the relation of conjectured bound and observed price of anarchy are on pages 172 to 178.
- Detailed tables for $p = 1$ are on pages 179 to 186.
- Tables giving more TRx values for $p \in \{1, 2, 3\}$ and $x \in \{30, 40, \ldots, 100\}$ are on pages 187 to 210.
- Plots comparing Perakis' bound to our bounds are on pages 212 to 215.

Further explanations are given in Ch. 3.

B.1 Hexagonal Binning Plots

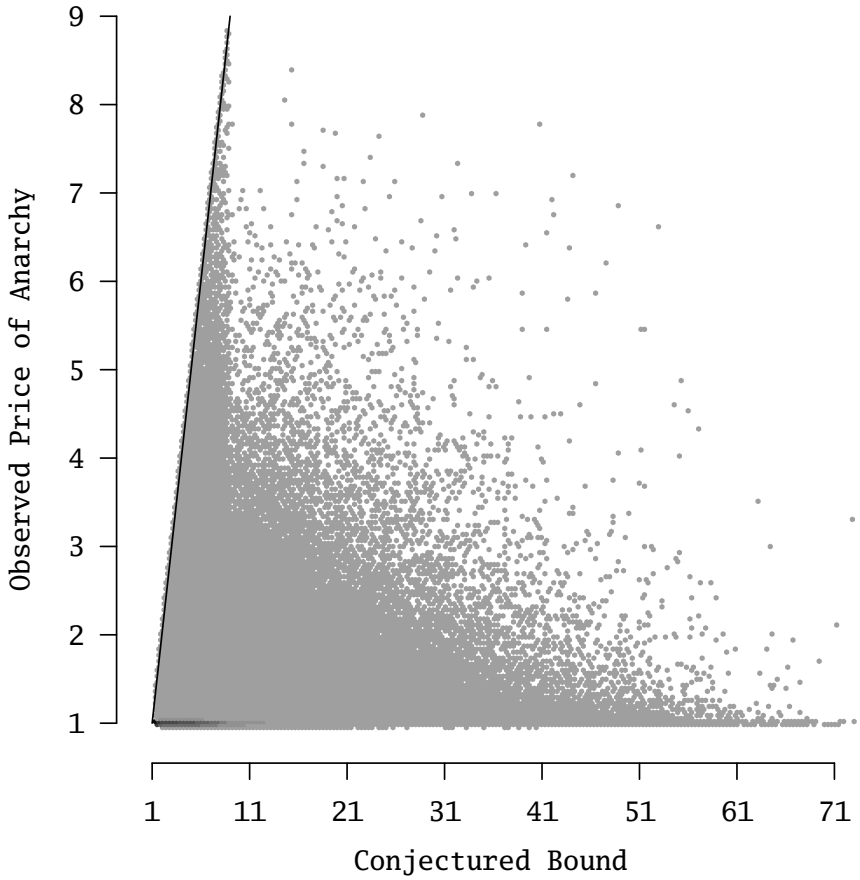


Figure B.1. Random instances with affine element latency functions, i.e., $p = 1$. All 2,293,023 instances from the series with *fixed consumption number range* are shown that yielded an acceptable solution in the binary search. The borderline on the left is barely visible, since it is covered with hexagons.

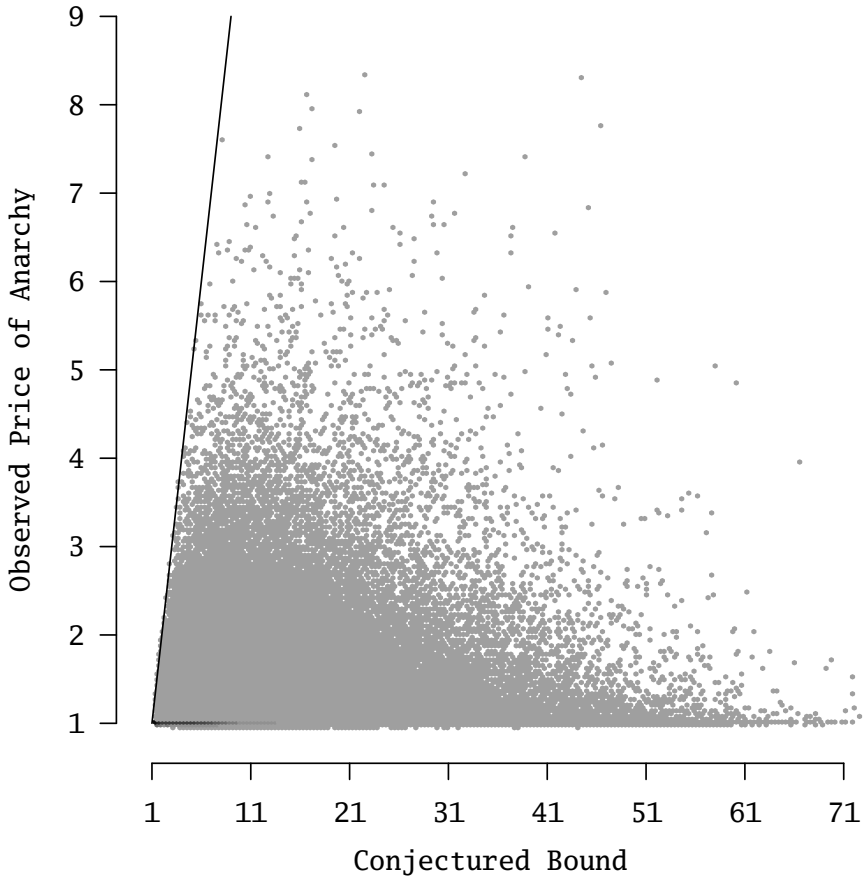


Figure B.2. Random instances with affine element latency functions, i.e., $p = 1$. All 2,294,639 instances from the series with *fixed relevance number range* are shown that yielded an acceptable solution in the binary search.

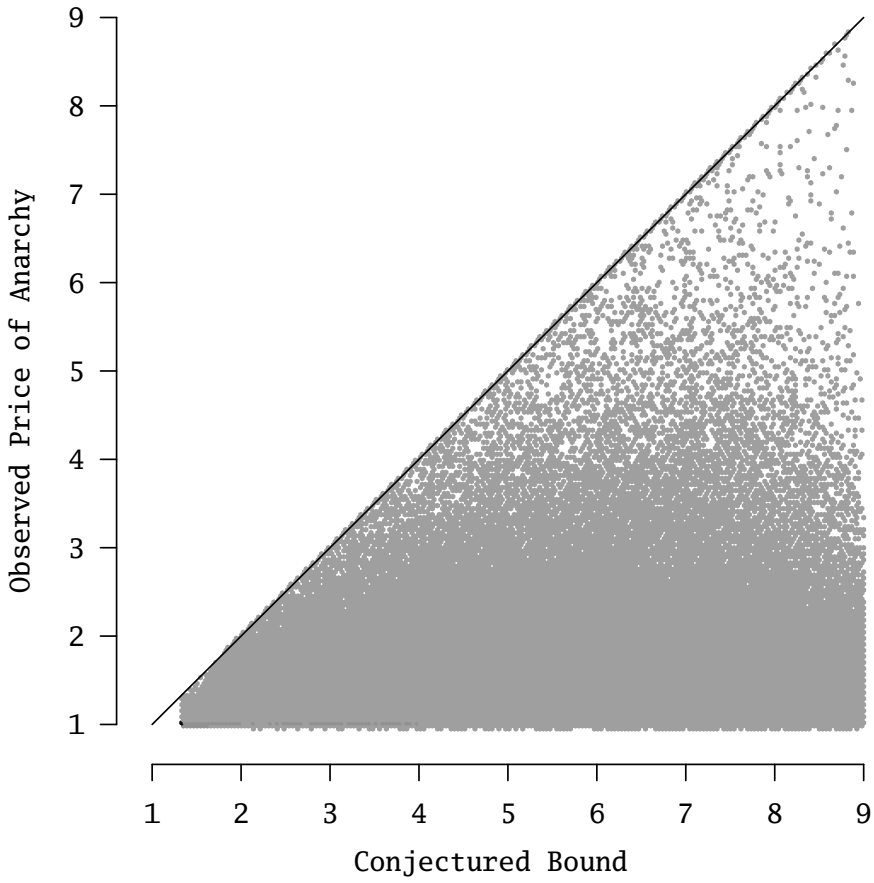


Figure B.3. Random instances with affine element latency functions, i.e., $p = 1$. The altogether 4,587,662 instances from Fig. B.1 and Fig. B.2 are considered. However, for the sake of a better scale, only those are shown which have conjectured bound at most the maximum observed price of anarchy, resulting in 3,556,306 instances shown.

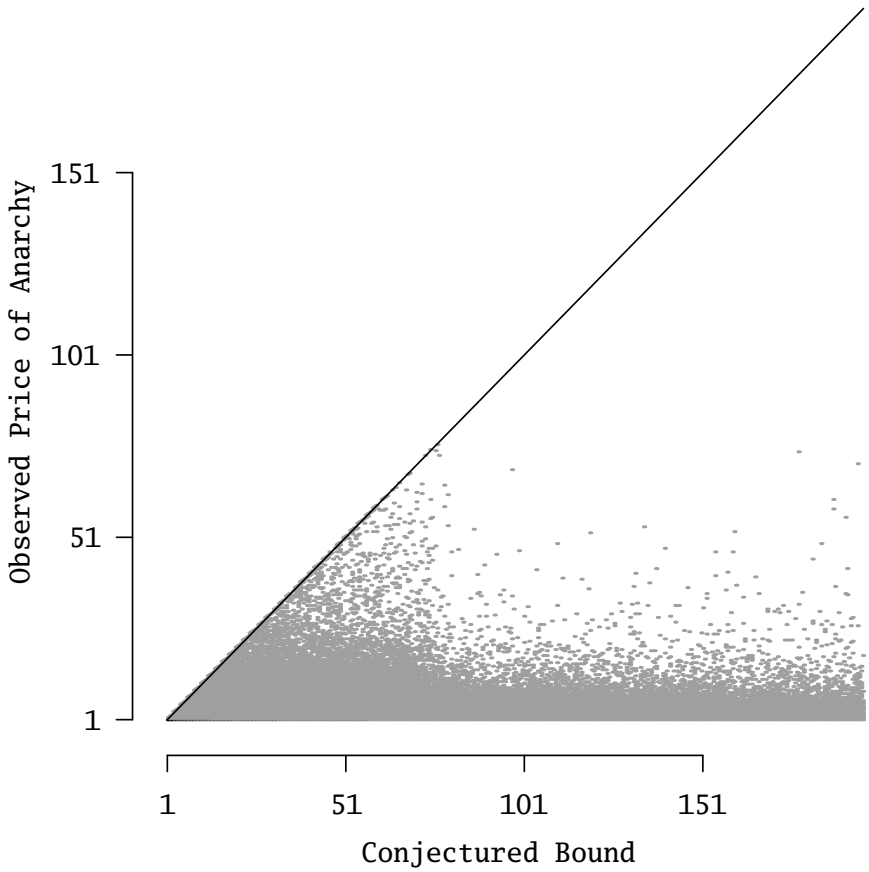


Figure B.4. Random instances with $p = 2$. All instances are considered that yielded an acceptable solution in the binary search. However, for the sake of a better scale, only those are shown which have conjectured bound smaller than the maximum observed price of anarchy, resulting in 4,030,513 instances shown.

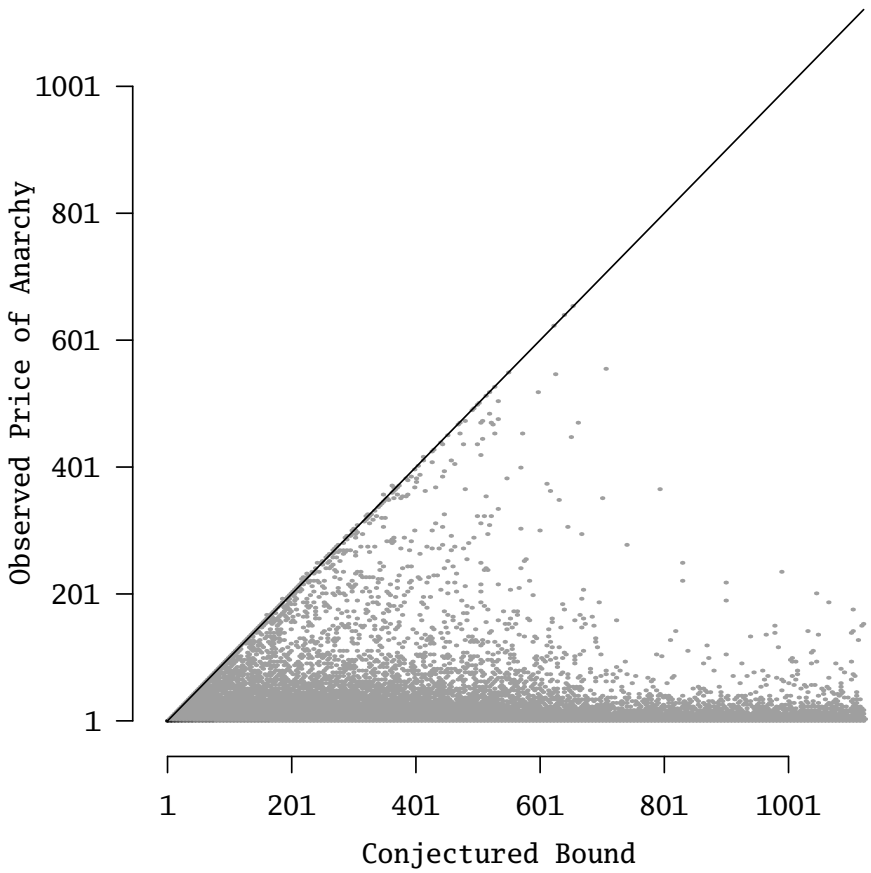


Figure B.5. Random instances with $p = 3$. All instances are considered that yielded an acceptable solution in the binary search. However, for the sake of a better scale, only those are shown which have conjectured bound smaller than the maximum observed price of anarchy, resulting in 1,787,304 instances shown. If watching the borderline closely, one notices slightly more overstepping than for $p \in \{1, 2\}$. We attribute these to numerical inaccuracies.

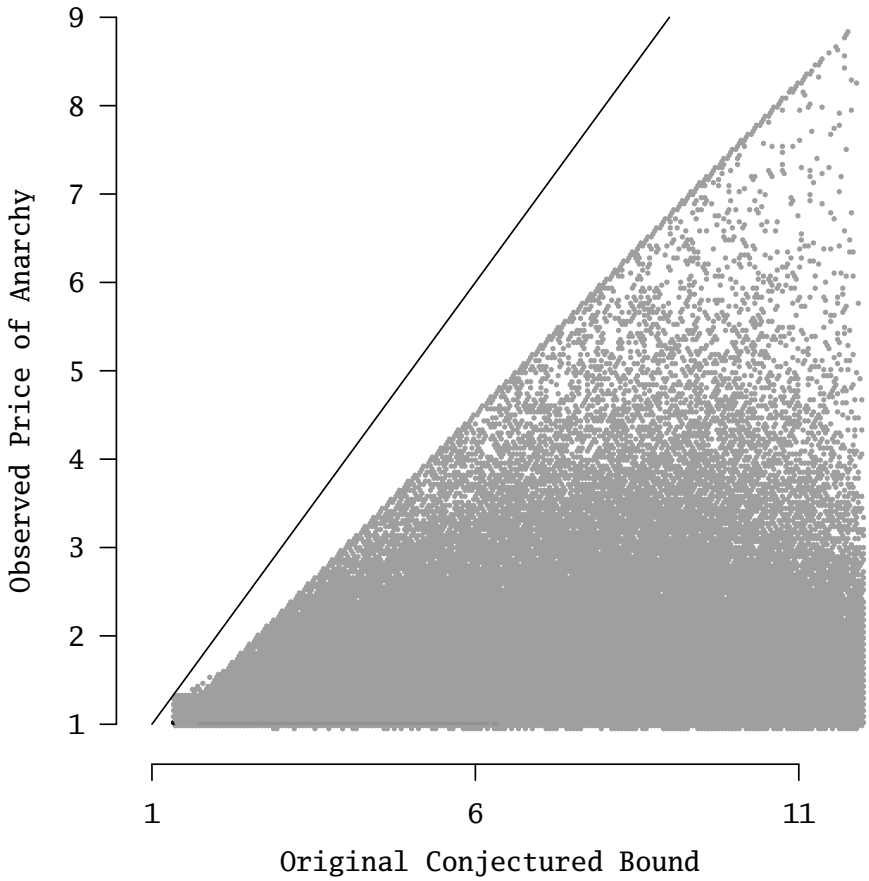


Figure B.6. These are the instances from Fig. B.3, i.e., with $p = 1$. We use an early version of our conjecture, which we call "original conjectured bound". It states a bound of $\frac{1}{1-\beta}\gamma^p$. It clearly is too pessimistic, except for $\gamma = 1$.

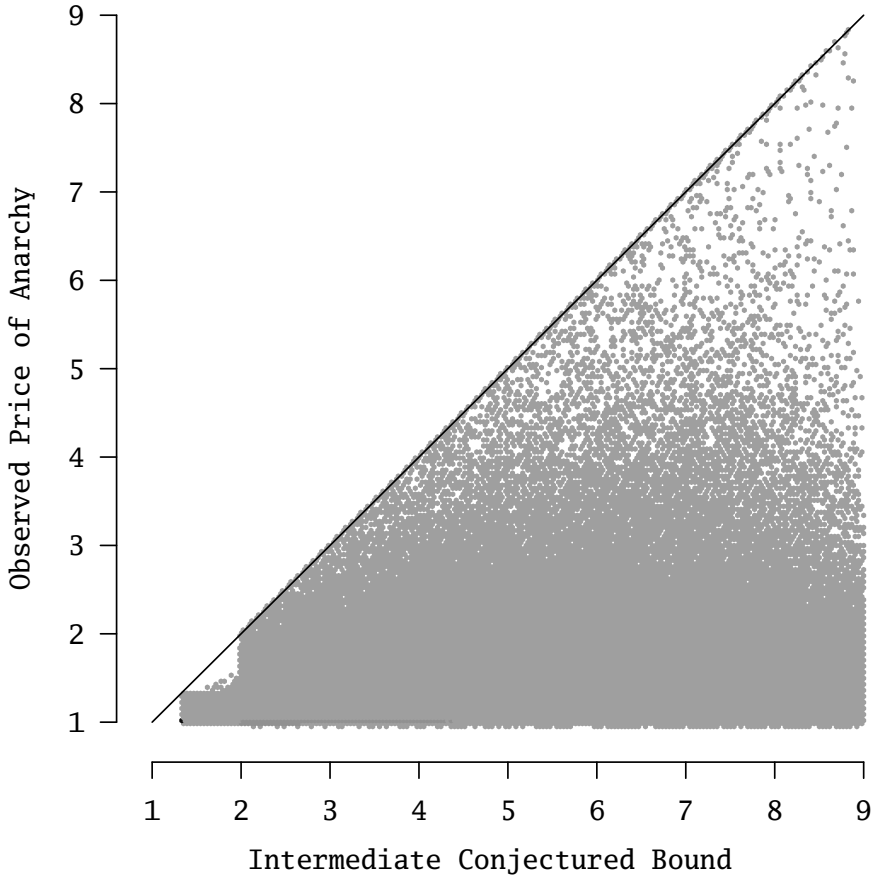


Figure B.7. Similar to the previous plot, but we use an improved conjecture. It is not yet the final one, so we call it "intermediate conjectured bound". It states a bound of $\frac{1}{1-\beta}\gamma^p$ for $\gamma \leq (1+p)^{\frac{1}{p}}$ (this threshold is 2 here) and γ^p for $\gamma > (1+p)^{\frac{1}{p}}$. It is still too pessimistic in the lower range of γ. Fig. B.3 on page 174 shows the currently final version of the conjecture.

B.2 Tables

B.2.1 Detailed Tables for $p = 1$

Table B.1. Single element latency functions, $d = 1$, fixed consumption number range.

n	$r_{\max}$	%C	%GC	%Perakis	%OK	avg.	#($\gamma > 1$)	%TR50	%TR90	%TR100
2	1.0	90.10	88.99	88.99	100.00	1.09	26331	51.79	1.82	0.1861
3	1.0	72.14	61.53	61.52	100.00	1.13	31504	28.74	1.55	0.1397
4	1.0	43.09	22.37	22.36	100.00	1.17	31982	15.10	0.97	0.1126
5	1.0	12.90	0.00	0.00	100.00	1.21	31999	9.41	0.62	0.0469
6	1.0	0.00	0.00	0.00	100.00	1.25	32000	6.97	0.40	0.0375
9	1.0	0.00	0.00	0.00	100.00	1.35	32000	3.90	0.12	0.0062
2	5.0	93.01	88.55	88.55	99.70	1.07	26108	38.06	0.04	0.0000
3	5.0	73.61	56.47	56.45	99.57	1.09	31371	16.02	0.04	0.0000
4	5.0	41.02	16.98	16.97	99.54	1.11	31834	5.24	0.00	0.0000
5	5.0	10.91	0.00	0.00	99.56	1.12	31860	1.56	0.01	0.0000
6	5.0	0.00	0.00	0.00	99.57	1.13	31862	0.52	0.01	0.0000
9	5.0	0.00	0.00	0.00	99.57	1.15	31863	0.04	0.00	0.0000
2	9.0	91.70	86.56	86.56	99.62	1.07	26322	33.25	0.06	0.0000
3	9.0	70.97	52.81	52.81	99.51	1.10	31333	13.16	0.04	0.0000
4	9.0	37.40	14.69	14.68	99.31	1.12	31768	3.89	0.01	0.0000
5	9.0	9.09	0.00	0.00	99.27	1.13	31766	0.95	0.00	0.0000
6	9.0	0.00	0.00	0.00	99.25	1.14	31759	0.29	0.00	0.0000
9	9.0	0.00	0.00	0.00	99.34	1.16	31788	0.01	0.00	0.0000

Table B.2. Single element latency functions, $d = 1$, fixed relevance number range.

n	$c_{\max}$	%C	%GC	%Perakis	%OK	avg.	$\#(\gamma > 1)$	%TR50	%TR90	%TR100
2	1.0	90.33	89.36	89.36	100.00	1.01	26242	48.08	0.00	0.0000
3	1.0	71.79	61.25	61.24	100.00	1.02	31512	23.74	0.00	0.0000
4	1.0	43.02	22.12	22.12	99.99	1.03	31980	9.70	0.00	0.0000
5	1.0	13.37	0.00	0.00	99.97	1.03	31989	3.57	0.00	0.0000
6	1.0	0.00	0.00	0.00	99.97	1.04	31990	1.13	0.00	0.0000
9	1.0	0.00	0.00	0.00	99.93	1.04	31979	0.06	0.00	0.0000
2	5.0	92.56	88.19	88.19	99.82	1.05	26335	38.09	0.03	0.0000
3	5.0	73.82	56.61	56.59	99.70	1.07	31430	15.99	0.03	0.0032
4	5.0	41.18	17.38	17.38	99.64	1.09	31865	4.91	0.00	0.0000
5	5.0	11.27	0.00	0.00	99.55	1.09	31856	1.44	0.00	0.0000
6	5.0	0.00	0.00	0.00	99.67	1.10	31893	0.40	0.00	0.0000
9	5.0	0.00	0.00	0.00	99.67	1.11	31894	0.01	0.00	0.0000
2	9.0	91.77	86.53	86.53	99.67	1.07	26260	33.53	0.04	0.0000
3	9.0	70.91	52.46	52.45	99.52	1.10	31353	13.17	0.03	0.0000
4	9.0	37.19	14.67	14.67	99.28	1.12	31755	3.80	0.00	0.0000
5	9.0	8.94	0.00	0.00	99.33	1.13	31787	1.04	0.00	0.0000
6	9.0	0.00	0.00	0.00	99.18	1.14	31739	0.29	0.00	0.0000
9	9.0	0.00	0.00	0.00	99.38	1.16	31800	0.03	0.00	0.0000

Table B.3. Mixed element latency functions, $d = 1$, fixed consumption number range.

n	$r_{\max}$	%C	%GC	%Perakis	%OK	avg.	$\#(\gamma > 1)$	%TR50	%TR90	%TR100
2	1.0	77.38	55.46	55.46	100.00	1.04	26511	48.35	0.72	0.0943
3	1.0	43.76	21.63	21.63	100.00	1.06	31545	26.08	1.01	0.1014
4	1.0	14.82	3.38	3.38	100.00	1.09	31980	12.42	0.85	0.1376
5	1.0	1.77	0.00	0.00	100.00	1.12	32000	7.08	0.76	0.1438
6	1.0	0.00	0.00	0.00	100.00	1.14	31999	4.59	0.58	0.0906
9	1.0	0.00	0.00	0.00	99.99	1.23	31998	3.29	0.47	0.0469
2	5.0	83.87	55.25	55.24	99.90	1.01	26323	35.82	0.02	0.0000
3	5.0	49.34	20.77	20.76	99.96	1.02	31498	15.27	0.05	0.0000
4	5.0	15.27	2.74	2.73	99.89	1.02	31946	4.92	0.02	0.0000
5	5.0	1.68	0.00	0.00	99.92	1.03	31975	1.38	0.01	0.0000
6	5.0	0.00	0.00	0.00	99.92	1.04	31976	0.55	0.02	0.0000
9	5.0	0.00	0.00	0.00	99.87	1.07	31958	0.15	0.00	0.0000
2	9.0	83.49	54.28	54.28	99.96	1.01	26450	31.04	0.01	0.0000
3	9.0	47.66	18.80	18.80	99.94	1.01	31476	11.94	0.03	0.0000
4	9.0	13.53	2.48	2.48	99.96	1.02	31968	3.40	0.01	0.0031
5	9.0	1.29	0.00	0.00	99.96	1.03	31987	1.02	0.01	0.0000
6	9.0	0.00	0.00	0.00	99.94	1.04	31982	0.31	0.00	0.0000
9	9.0	0.00	0.00	0.00	99.92	1.06	31973	0.05	0.00	0.0000

Table B.4. Mixed element latency functions, $d = 1$, fixed relevance number range.

n	$c_{\max}$	%C	%GC	%Perakis	%OK	avg.	$\#(\gamma > 1)$	%TR50	%TR90	%TR100
2	1.0	77.60	55.47	55.46	100.00	1.00	26372	46.16	0.00	0.0000
3	1.0	43.88	21.55	21.55	100.00	1.00	31504	23.36	0.01	0.0000
4	1.0	14.13	3.44	3.44	100.00	1.01	31978	9.83	0.00	0.0000
5	1.0	1.86	0.00	0.00	100.00	1.01	32000	3.47	0.00	0.0000
6	1.0	0.00	0.00	0.00	99.99	1.01	31998	1.18	0.00	0.0000
9	1.0	0.00	0.00	0.00	100.00	1.01	31999	0.04	0.00	0.0000
2	5.0	84.36	55.87	55.86	99.98	1.00	26451	35.46	0.01	0.0000
3	5.0	49.10	20.24	20.24	99.98	1.01	31477	14.60	0.03	0.0000
4	5.0	14.87	2.72	2.72	99.98	1.01	31974	4.74	0.02	0.0000
5	5.0	1.49	0.00	0.00	99.97	1.02	31989	1.32	0.00	0.0000
6	5.0	0.00	0.00	0.00	99.97	1.02	31990	0.45	0.00	0.0000
9	5.0	0.00	0.00	0.00	99.96	1.04	31987	0.04	0.00	0.0000
2	9.0	83.50	54.82	54.81	99.93	1.01	26425	31.32	0.02	0.0000
3	9.0	47.10	18.87	18.87	99.94	1.01	31523	11.90	0.03	0.0000
4	9.0	13.43	2.44	2.44	99.95	1.02	31967	3.43	0.01	0.0000
5	9.0	1.29	0.00	0.00	99.93	1.03	31977	0.99	0.02	0.0000
6	9.0	0.00	0.00	0.00	99.95	1.04	31985	0.25	0.01	0.0000
9	9.0	0.00	0.00	0.00	99.90	1.06	31967	0.04	0.00	0.0000

Table B.5. Single element latency functions, $d = 10$, fixed consumption number range.

n	$r_{\max}$	%C	%GC	%Perakis	%OK	avg.	#($\gamma > 1$)	%TR50	%TR90	%TR100
2	1.0	90.50	89.46	89.46	100.00	1.09	26210	51.85	1.84	0.0229
3	1.0	71.85	61.02	61.02	99.99	1.13	31486	28.34	1.52	0.0032
4	1.0	43.54	22.01	22.01	99.99	1.17	31982	15.45	0.92	0.0125
5	1.0	13.60	0.00	0.00	99.97	1.20	31990	9.03	0.57	0.0000
6	1.0	0.00	0.00	0.00	99.90	1.24	31967	6.66	0.36	0.0000
9	1.0	0.00	0.00	0.00	99.97	1.35	31991	3.84	0.08	0.0000
2	5.0	92.57	88.15	88.14	99.38	1.06	26269	37.93	0.05	0.0000
3	5.0	74.32	56.79	56.79	98.91	1.08	31132	15.72	0.01	0.0000
4	5.0	41.23	17.32	17.31	98.54	1.10	31517	4.81	0.00	0.0000
5	5.0	10.67	0.00	0.00	98.26	1.11	31442	1.63	0.00	0.0000
6	5.0	0.00	0.00	0.00	98.23	1.12	31433	0.40	0.00	0.0000
9	5.0	0.00	0.00	0.00	98.27	1.14	31445	0.06	0.00	0.0000
2	9.0	91.76	86.29	86.29	99.43	1.07	26306	32.98	0.03	0.0000
3	9.0	70.53	52.11	52.10	98.58	1.09	31049	13.22	0.02	0.0000
4	9.0	37.04	14.65	14.64	98.28	1.10	31438	3.87	0.00	0.0000
5	9.0	9.53	0.00	0.00	98.06	1.11	31380	1.03	0.00	0.0000
6	9.0	0.00	0.00	0.00	97.94	1.12	31340	0.25	0.00	0.0000
9	9.0	0.00	0.00	0.00	97.75	1.15	31279	0.01	0.00	0.0000

Table B.6. Single element latency functions, $d = 10$, fixed relevance number range.

n	$c_{\max}$	%C	%GC	%Perakis	%OK	avg.	$\#(\gamma > 1)$	%TR50	%TR90	%TR100
2	1.0	90.29	89.27	89.27	100.00	1.01	26530	47.57	0.00	0.0000
3	1.0	71.96	61.52	61.52	99.98	1.02	31471	24.16	0.00	0.0000
4	1.0	42.50	22.23	22.22	99.92	1.03	31955	9.80	0.00	0.0000
5	1.0	13.22	0.00	0.00	99.90	1.03	31968	3.31	0.00	0.0000
6	1.0	0.00	0.00	0.00	99.92	1.04	31973	1.14	0.00	0.0000
9	1.0	0.00	0.00	0.00	99.91	1.04	31972	0.04	0.00	0.0000
2	5.0	92.80	88.53	88.52	99.54	1.05	26224	37.36	0.03	0.0000
3	5.0	73.76	56.34	56.33	99.17	1.07	31244	15.47	0.02	0.0000
4	5.0	41.26	17.29	17.28	99.02	1.08	31662	5.00	0.00	0.0000
5	5.0	11.24	0.00	0.00	98.86	1.08	31634	1.47	0.00	0.0000
6	5.0	0.00	0.00	0.00	98.82	1.09	31622	0.40	0.00	0.0000
9	5.0	0.00	0.00	0.00	98.80	1.11	31617	0.01	0.00	0.0000
2	9.0	91.50	86.26	86.26	99.28	1.06	26127	33.44	0.02	0.0000
3	9.0	70.90	52.46	52.45	98.72	1.09	31112	13.05	0.01	0.0000
4	9.0	37.33	14.70	14.69	98.28	1.11	31436	3.83	0.00	0.0000
5	9.0	9.16	0.00	0.00	98.09	1.11	31388	1.01	0.00	0.0000
6	9.0	0.00	0.00	0.00	97.98	1.12	31353	0.26	0.00	0.0000
9	9.0	0.00	0.00	0.00	97.74	1.14	31276	0.02	0.00	0.0000

Table B.7. Mixed element latency functions, $d = 10$, fixed consumption number range.

n	$r_{\max}$	%C	%GC	%Perakis	%OK	avg.	#($\gamma > 1$)	%TR50	%TR90	%TR100
2	1.0	77.61	55.93	55.93	100.00	1.10	26298	52.99	0.96	0.0114
3	1.0	44.25	21.95	21.95	99.99	1.12	31501	28.55	0.77	0.0032
4	1.0	14.79	3.51	3.51	100.00	1.14	31979	14.13	0.58	0.0094
5	1.0	1.75	0.00	0.00	99.99	1.14	31997	7.00	0.33	0.0031
6	1.0	0.00	0.00	0.00	99.94	1.16	31982	4.28	0.24	0.0031
9	1.0	0.00	0.00	0.00	99.99	1.18	31996	1.72	0.12	0.0031
2	5.0	84.36	55.55	55.55	99.46	1.03	26131	36.85	0.03	0.0000
3	5.0	48.97	20.59	20.58	99.34	1.04	31306	15.71	0.06	0.0000
4	5.0	14.97	2.84	2.84	99.19	1.05	31726	5.31	0.01	0.0000
5	5.0	1.45	0.00	0.00	99.15	1.06	31728	1.53	0.02	0.0000
6	5.0	0.00	0.00	0.00	99.18	1.06	31738	0.45	0.00	0.0000
9	5.0	0.00	0.00	0.00	99.01	1.08	31683	0.04	0.00	0.0000
2	9.0	83.10	54.33	54.33	99.60	1.02	26359	31.68	0.04	0.0000
3	9.0	46.82	18.47	18.46	99.48	1.03	31326	12.52	0.03	0.0000
4	9.0	13.59	2.30	2.29	99.28	1.04	31751	3.97	0.01	0.0000
5	9.0	1.31	0.00	0.00	99.28	1.05	31768	0.96	0.00	0.0000
6	9.0	0.00	0.00	0.00	99.30	1.05	31775	0.25	0.00	0.0000
9	9.0	0.00	0.00	0.00	99.16	1.07	31730	0.01	0.00	0.0000

Table B.8. Mixed element latency functions, $d = 10$, fixed relevance number range.

n	$c_{\max}$	%C	%GC	%Perakis	%OK	avg.	$\#(\gamma > 1)$	%TR50	%TR90	%TR100
2	1.0	77.73	55.53	55.52	100.00	1.01	26317	46.55	0.00	0.0000
3	1.0	44.38	21.86	21.86	99.98	1.02	31490	24.51	0.09	0.0000
4	1.0	14.90	3.71	3.71	99.95	1.03	31957	10.22	0.01	0.0000
5	1.0	1.86	0.00	0.00	99.95	1.03	31984	3.91	0.00	0.0000
6	1.0	0.00	0.00	0.00	99.95	1.03	31984	1.50	0.00	0.0000
9	1.0	0.00	0.00	0.00	99.95	1.04	31984	0.09	0.00	0.0000
2	5.0	84.28	55.63	55.63	99.72	1.02	26283	35.84	0.03	0.0000
3	5.0	49.16	20.12	20.11	99.52	1.03	31345	15.01	0.04	0.0000
4	5.0	15.04	2.84	2.84	99.53	1.04	31835	5.01	0.01	0.0000
5	5.0	1.49	0.00	0.00	99.49	1.04	31836	1.47	0.00	0.0000
6	5.0	0.00	0.00	0.00	99.49	1.05	31838	0.41	0.00	0.0000
9	5.0	0.00	0.00	0.00	99.50	1.06	31839	0.02	0.00	0.0000
2	9.0	83.46	54.38	54.38	99.60	1.02	26323	32.02	0.02	0.0000
3	9.0	47.39	18.79	18.78	99.41	1.03	31326	12.31	0.03	0.0000
4	9.0	13.53	2.41	2.41	99.36	1.04	31779	3.84	0.02	0.0000
5	9.0	1.33	0.00	0.00	99.30	1.05	31777	1.00	0.01	0.0000
6	9.0	0.00	0.00	0.00	99.25	1.06	31761	0.31	0.00	0.0000
9	9.0	0.00	0.00	0.00	99.17	1.07	31736	0.03	0.00	0.0000

B.2.2 TRx Values for $p \in \{1, 2, 3\}$

Table B.9. TRx values for degree $p = 1$, demand $d = 1$, single element latency functions, and fixed consumption number range.

n	$r_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	100.00	26331	81.31	66.34	51.79	36.77	15.98	2.76	1.82	0.1861
3	1.0	100.00	31504	63.63	43.37	28.74	17.30	7.51	2.64	1.55	0.1397
4	1.0	100.00	31982	46.38	26.34	15.10	8.23	3.90	1.67	0.97	0.1126
5	1.0	100.00	31999	35.08	17.87	9.41	4.97	2.32	1.03	0.62	0.0469
6	1.0	100.00	32000	28.42	13.47	6.97	3.57	1.65	0.71	0.40	0.0375
9	1.0	100.00	32000	20.41	9.00	3.90	1.52	0.57	0.27	0.12	0.0062
2	5.0	99.70	26108	68.24	51.85	38.06	24.80	8.63	0.14	0.04	0.0000
3	5.0	99.57	31371	44.43	27.17	16.02	8.13	2.20	0.11	0.04	0.0000
4	5.0	99.54	31834	24.90	11.48	5.24	2.07	0.43	0.02	0.00	0.0000
5	5.0	99.56	31860	13.55	4.71	1.56	0.42	0.07	0.02	0.01	0.0000
6	5.0	99.57	31862	7.15	1.97	0.52	0.10	0.03	0.01	0.01	0.0000
9	5.0	99.57	31863	1.26	0.21	0.04	0.01	0.00	0.00	0.00	0.0000
2	9.0	99.62	26322	61.29	46.14	33.25	21.83	7.53	0.17	0.06	0.0000
3	9.0	99.51	31333	36.59	22.04	13.16	6.81	1.96	0.09	0.04	0.0000
4	9.0	99.31	31768	18.41	8.35	3.89	1.67	0.35	0.02	0.01	0.0000
5	9.0	99.27	31766	8.37	2.84	0.95	0.32	0.06	0.01	0.00	0.0000
6	9.0	99.25	31759	4.24	1.10	0.29	0.06	0.00	0.00	0.00	0.0000
9	9.0	99.34	31788	0.51	0.07	0.01	0.00	0.00	0.00	0.00	0.0000

Table B.10. TRx values for degree $p = 1$, demand $d = 1$, single element latency functions, and fixed relevance number range.

n	$c_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	100.00	26242	77.17	61.95	48.08	33.10	12.13	0.00	0.00	0.0000
3	1.0	100.00	31512	57.80	37.98	23.74	12.41	3.11	0.00	0.00	0.0000
4	1.0	99.99	31980	39.89	20.31	9.70	3.77	0.53	0.00	0.00	0.0000
5	1.0	99.97	31989	26.15	10.13	3.57	0.88	0.06	0.00	0.00	0.0000
6	1.0	99.97	31990	16.08	4.45	1.13	0.17	0.00	0.00	0.00	0.0000
9	1.0	99.93	31979	3.99	0.53	0.06	0.00	0.00	0.00	0.00	0.0000
2	5.0	99.82	26335	67.67	51.51	38.09	24.53	8.64	0.13	0.03	0.0000
3	5.0	99.70	31430	43.99	26.90	15.99	8.23	2.10	0.09	0.03	0.0032
4	5.0	99.64	31865	24.32	11.25	4.91	1.87	0.38	0.02	0.00	0.0000
5	5.0	99.55	31856	12.97	4.33	1.44	0.42	0.06	0.01	0.00	0.0000
6	5.0	99.67	31893	6.56	1.73	0.40	0.09	0.02	0.00	0.00	0.0000
9	5.0	99.67	31894	0.90	0.11	0.01	0.01	0.00	0.00	0.00	0.0000
2	9.0	99.67	26260	61.33	45.89	33.53	21.80	7.66	0.15	0.04	0.0000
3	9.0	99.52	31353	36.52	22.02	13.17	6.79	1.76	0.07	0.03	0.0000
4	9.0	99.28	31755	18.46	8.38	3.80	1.49	0.30	0.02	0.00	0.0000
5	9.0	99.33	31787	8.71	2.93	1.04	0.31	0.05	0.02	0.00	0.0000
6	9.0	99.18	31739	4.03	1.14	0.29	0.10	0.03	0.01	0.00	0.0000
9	9.0	99.38	31800	0.57	0.12	0.03	0.01	0.00	0.00	0.00	0.0000

Table B.11. TRx values for degree $p = 1$, demand $d = 1$, mixed element latency functions, and fixed consumption number range.

n	$r_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	100.00	26511	77.99	62.60	48.35	33.67	13.88	1.27	0.72	0.0943
3	1.0	100.00	31545	59.16	39.57	26.08	15.26	5.91	1.92	1.01	0.1014
4	1.0	100.00	31980	42.69	23.10	12.42	6.22	2.82	1.44	0.85	0.1376
5	1.0	100.00	32000	29.94	14.07	7.08	3.64	1.92	1.22	0.76	0.1438
6	1.0	100.00	31999	22.55	9.56	4.59	2.43	1.50	0.93	0.58	0.0906
9	1.0	99.99	31998	12.76	5.91	3.29	2.08	1.38	0.83	0.47	0.0469
2	5.0	99.90	26323	64.50	48.71	35.82	23.24	7.92	0.07	0.02	0.0000
3	5.0	99.96	31498	41.14	25.29	15.27	7.73	2.14	0.15	0.05	0.0000
4	5.0	99.89	31946	22.20	10.52	4.92	1.96	0.44	0.07	0.02	0.0000
5	5.0	99.92	31975	11.34	4.04	1.38	0.52	0.13	0.03	0.01	0.0000
6	5.0	99.92	31976	6.02	1.75	0.55	0.18	0.10	0.03	0.02	0.0000
9	5.0	99.87	31958	1.18	0.35	0.15	0.06	0.02	0.02	0.00	0.0000
2	9.0	99.96	26450	57.05	42.52	31.04	19.70	6.80	0.03	0.01	0.0000
3	9.0	99.94	31476	33.20	19.97	11.94	6.10	1.69	0.11	0.03	0.0000
4	9.0	99.96	31968	15.97	7.21	3.40	1.41	0.33	0.04	0.01	0.0031
5	9.0	99.96	31987	7.24	2.58	1.02	0.34	0.07	0.03	0.01	0.0000
6	9.0	99.94	31982	3.20	0.98	0.31	0.11	0.04	0.01	0.00	0.0000
9	9.0	99.92	31973	0.46	0.15	0.05	0.03	0.01	0.00	0.00	0.0000

Table B.12. TRx values for degree $p = 1$, demand $d = 1$, mixed element latency functions, and fixed relevance number range.

n	$c_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	100.00	26372	75.92	60.22	46.16	31.23	11.68	0.01	0.00	0.0000
3	1.0	100.00	31504	56.55	36.97	23.36	12.38	2.99	0.08	0.01	0.0000
4	1.0	100.00	31978	38.35	19.50	9.83	3.94	0.65	0.03	0.00	0.0000
5	1.0	100.00	32000	24.08	9.29	3.47	0.93	0.09	0.00	0.00	0.0000
6	1.0	99.99	31998	15.44	4.44	1.18	0.20	0.01	0.00	0.00	0.0000
9	1.0	100.00	31999	3.34	0.37	0.04	0.00	0.00	0.00	0.00	0.0000
2	5.0	99.98	26451	63.64	48.11	35.46	22.73	7.91	0.03	0.01	0.0000
3	5.0	99.98	31477	39.98	24.26	14.60	7.52	1.99	0.10	0.03	0.0000
4	5.0	99.98	31974	21.88	10.13	4.74	1.90	0.39	0.05	0.02	0.0000
5	5.0	99.97	31989	10.75	3.72	1.32	0.38	0.07	0.02	0.00	0.0000
6	5.0	99.97	31990	5.46	1.47	0.45	0.12	0.03	0.01	0.00	0.0000
9	5.0	99.96	31987	0.79	0.14	0.04	0.02	0.01	0.00	0.00	0.0000
2	9.0	99.93	26425	57.66	43.04	31.32	20.30	7.01	0.06	0.02	0.0000
3	9.0	99.94	31523	32.96	20.08	11.90	6.25	1.81	0.10	0.03	0.0000
4	9.0	99.95	31967	15.87	7.42	3.43	1.33	0.27	0.03	0.01	0.0000
5	9.0	99.93	31977	7.42	2.63	0.99	0.41	0.12	0.04	0.02	0.0000
6	9.0	99.95	31985	3.31	0.88	0.25	0.07	0.03	0.01	0.01	0.0000
9	9.0	99.90	31967	0.51	0.16	0.04	0.01	0.00	0.00	0.00	0.0000

Table B.13. TRx values for degree $p = 1$, demand $d = 10$, single element latency functions, and fixed consumption number range.

n	$r_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	100.00	26210	81.40	66.24	51.85	36.45	15.60	2.69	1.84	0.0229
3	1.0	99.99	31486	62.71	43.02	28.34	17.15	7.30	2.70	1.52	0.0032
4	1.0	99.99	31982	46.39	26.87	15.45	8.56	4.11	1.68	0.92	0.0125
5	1.0	99.97	31990	34.56	17.41	9.03	4.79	2.30	0.98	0.57	0.0000
6	1.0	99.90	31967	28.07	13.29	6.66	3.33	1.54	0.67	0.36	0.0000
9	1.0	99.97	31991	20.54	8.96	3.84	1.42	0.50	0.18	0.08	0.0000
2	5.0	99.38	26269	68.38	52.18	37.93	24.47	8.38	0.13	0.05	0.0000
3	5.0	98.91	31132	45.01	26.78	15.72	8.23	2.25	0.04	0.01	0.0000
4	5.0	98.54	31517	24.55	11.04	4.81	1.79	0.34	0.03	0.00	0.0000
5	5.0	98.26	31442	13.05	4.53	1.63	0.46	0.06	0.01	0.00	0.0000
6	5.0	98.23	31433	6.78	1.69	0.40	0.06	0.01	0.00	0.00	0.0000
9	5.0	98.27	31445	1.28	0.24	0.06	0.01	0.00	0.00	0.00	0.0000
2	9.0	99.43	26306	60.47	45.27	32.98	21.36	7.46	0.11	0.03	0.0000
3	9.0	98.58	31049	36.24	22.10	13.22	6.75	1.74	0.05	0.02	0.0000
4	9.0	98.28	31438	18.31	8.44	3.87	1.58	0.30	0.01	0.00	0.0000
5	9.0	98.06	31380	8.60	2.94	1.03	0.28	0.02	0.00	0.00	0.0000
6	9.0	97.94	31340	3.68	0.94	0.25	0.06	0.01	0.00	0.00	0.0000
9	9.0	97.75	31279	0.45	0.07	0.01	0.00	0.00	0.00	0.00	0.0000

Table B.14. TRx values for degree $p = 1$, demand $d = 10$, single element latency functions, and fixed relevance number range.

n	$c_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	100.00	26530	76.94	61.59	47.57	32.45	12.02	0.00	0.00	0.0000
3	1.0	99.98	31471	58.08	38.10	24.16	12.80	3.02	0.00	0.00	0.0000
4	1.0	99.92	31955	40.11	20.38	9.80	3.78	0.55	0.00	0.00	0.0000
5	1.0	99.90	31968	25.46	9.57	3.31	0.78	0.03	0.00	0.00	0.0000
6	1.0	99.92	31973	16.70	4.54	1.14	0.17	0.00	0.00	0.00	0.0000
9	1.0	99.91	31972	4.22	0.47	0.04	0.01	0.00	0.00	0.00	0.0000
2	5.0	99.54	26224	67.39	51.59	37.36	23.99	8.24	0.11	0.03	0.0000
3	5.0	99.17	31244	43.78	26.56	15.47	7.76	2.09	0.03	0.02	0.0000
4	5.0	99.02	31662	24.05	10.96	5.00	1.98	0.36	0.01	0.00	0.0000
5	5.0	98.86	31634	12.95	4.46	1.47	0.46	0.04	0.00	0.00	0.0000
6	5.0	98.82	31622	6.58	1.70	0.40	0.06	0.01	0.00	0.00	0.0000
9	5.0	98.80	31617	0.93	0.11	0.01	0.00	0.00	0.00	0.00	0.0000
2	9.0	99.28	26127	61.02	45.62	33.44	21.71	7.48	0.09	0.02	0.0000
3	9.0	98.72	31112	36.68	22.12	13.05	6.74	1.87	0.04	0.01	0.0000
4	9.0	98.28	31436	17.74	8.11	3.83	1.51	0.25	0.02	0.00	0.0000
5	9.0	98.09	31388	8.55	2.91	1.01	0.25	0.03	0.00	0.00	0.0000
6	9.0	97.98	31353	3.95	0.99	0.26	0.07	0.01	0.01	0.00	0.0000
9	9.0	97.74	31276	0.43	0.08	0.02	0.00	0.00	0.00	0.00	0.0000

Table B.15. TRx values for degree $p = 1$, demand $d = 10$, mixed element latency functions, and fixed consumption number range.

n	$r_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	100.00	26298	81.33	66.95	52.99	37.25	16.35	2.51	0.96	0.0114
3	1.0	99.99	31501	62.54	42.67	28.55	16.77	6.79	1.87	0.77	0.0032
4	1.0	100.00	31979	45.23	25.27	14.13	6.98	2.96	1.19	0.58	0.0094
5	1.0	99.99	31997	31.92	15.00	7.00	3.15	1.40	0.65	0.33	0.0031
6	1.0	99.94	31982	23.62	9.54	4.28	1.93	0.90	0.44	0.24	0.0031
9	1.0	99.99	31996	11.07	3.89	1.72	0.79	0.43	0.21	0.12	0.0031
2	5.0	99.46	26131	65.98	50.20	36.85	23.88	8.97	0.21	0.03	0.0000
3	5.0	99.34	31306	42.40	26.07	15.71	8.49	2.73	0.37	0.06	0.0000
4	5.0	99.19	31726	23.62	11.06	5.31	2.22	0.62	0.11	0.01	0.0000
5	5.0	99.15	31728	12.65	4.53	1.53	0.52	0.09	0.02	0.02	0.0000
6	5.0	99.18	31738	6.06	1.66	0.45	0.13	0.04	0.02	0.00	0.0000
9	5.0	99.01	31683	0.95	0.19	0.04	0.02	0.00	0.00	0.00	0.0000
2	9.0	99.60	26359	57.86	43.71	31.68	20.51	7.56	0.18	0.04	0.0000
3	9.0	99.48	31326	34.24	20.74	12.52	6.83	2.19	0.25	0.03	0.0000
4	9.0	99.28	31751	16.88	8.11	3.97	1.63	0.43	0.06	0.01	0.0000
5	9.0	99.28	31768	7.48	2.63	0.96	0.29	0.05	0.01	0.00	0.0000
6	9.0	99.30	31775	3.54	0.90	0.25	0.06	0.01	0.00	0.00	0.0000
9	9.0	99.16	31730	0.36	0.05	0.01	0.01	0.00	0.00	0.00	0.0000

Table B.16. TRx values for degree $p = 1$, demand $d = 10$, mixed element latency functions, and fixed relevance number range.

n	$c_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	100.00	26317	76.13	60.62	46.55	31.63	12.04	0.09	0.00	0.0000
3	1.0	99.98	31490	57.14	38.16	24.51	13.71	4.50	0.53	0.09	0.0000
4	1.0	99.95	31957	39.18	20.43	10.22	4.26	1.03	0.15	0.01	0.0000
5	1.0	99.95	31984	26.20	10.29	3.91	1.28	0.26	0.03	0.00	0.0000
6	1.0	99.95	31984	16.83	4.92	1.50	0.34	0.06	0.01	0.00	0.0000
9	1.0	99.95	31984	4.43	0.65	0.09	0.01	0.00	0.00	0.00	0.0000
2	5.0	99.72	26283	64.48	49.11	35.84	22.91	8.29	0.19	0.03	0.0000
3	5.0	99.52	31345	41.35	25.30	15.01	7.98	2.51	0.30	0.04	0.0000
4	5.0	99.53	31835	23.07	10.69	5.01	2.04	0.48	0.08	0.01	0.0000
5	5.0	99.49	31836	11.63	4.11	1.47	0.45	0.10	0.02	0.00	0.0000
6	5.0	99.49	31838	6.05	1.62	0.41	0.08	0.01	0.00	0.00	0.0000
9	5.0	99.50	31839	0.73	0.08	0.02	0.01	0.00	0.00	0.00	0.0000
2	9.0	99.60	26323	58.09	43.46	32.02	20.90	7.41	0.17	0.02	0.0000
3	9.0	99.41	31326	34.04	20.57	12.31	6.66	2.11	0.23	0.03	0.0000
4	9.0	99.36	31779	16.89	8.00	3.84	1.67	0.48	0.09	0.02	0.0000
5	9.0	99.30	31777	7.87	2.75	1.00	0.35	0.08	0.02	0.01	0.0000
6	9.0	99.25	31761	3.36	0.93	0.31	0.09	0.02	0.01	0.00	0.0000
9	9.0	99.17	31736	0.45	0.10	0.03	0.01	0.00	0.00	0.00	0.0000

Table B.17. TRx values for degree $p = 2$, demand $d = 1$, single element latency functions, and fixed consumption number range.

n	$r_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	98.13	25765	50.11	39.23	26.95	8.18	3.32	2.42	1.79	0.5007
3	1.0	97.87	30806	25.67	17.26	10.14	4.40	2.15	1.39	1.01	0.2986
4	1.0	98.05	31362	13.07	7.89	4.57	2.33	1.21	0.84	0.65	0.2264
5	1.0	97.73	31273	7.27	3.75	2.02	1.05	0.63	0.42	0.32	0.0927
6	1.0	97.57	31222	4.99	2.45	1.22	0.68	0.42	0.29	0.19	0.0801
9	1.0	97.23	31114	2.34	0.91	0.40	0.20	0.14	0.08	0.04	0.0129
2	5.0	94.55	24637	33.99	24.33	13.88	2.02	0.23	0.10	0.04	0.0000
3	5.0	95.20	29974	13.26	7.85	3.49	0.41	0.05	0.03	0.00	0.0000
4	5.0	96.34	30815	3.79	1.72	0.65	0.11	0.02	0.01	0.00	0.0000
5	5.0	96.99	31038	1.01	0.40	0.10	0.01	0.00	0.00	0.00	0.0000
6	5.0	97.17	31095	0.32	0.08	0.03	0.00	0.00	0.00	0.00	0.0000
9	5.0	98.01	31364	0.03	0.01	0.00	0.00	0.00	0.00	0.00	0.0000
2	9.0	94.40	24661	29.87	21.00	11.72	1.71	0.14	0.06	0.02	0.0041
3	9.0	94.67	29814	10.99	6.67	3.20	0.39	0.05	0.03	0.01	0.0000
4	9.0	95.52	30550	3.01	1.33	0.53	0.07	0.01	0.00	0.00	0.0000
5	9.0	95.69	30622	0.70	0.22	0.06	0.00	0.00	0.00	0.00	0.0000
6	9.0	96.08	30745	0.14	0.05	0.01	0.00	0.00	0.00	0.00	0.0000
9	9.0	97.17	31095	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.0000

Table B.18. TRx values for degree $p = 2$, demand $d = 1$, single element latency functions, and fixed relevance number range.

n	$c_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	100.00	26227	41.61	30.88	18.18	1.99	0.00	0.00	0.00	0.0000
3	1.0	99.38	31302	19.08	11.61	5.21	0.39	0.00	0.00	0.00	0.0000
4	1.0	99.60	31860	6.83	3.27	1.10	0.05	0.00	0.00	0.00	0.0000
5	1.0	99.80	31937	2.15	0.70	0.15	0.00	0.00	0.00	0.00	0.0000
6	1.0	99.87	31957	0.60	0.18	0.01	0.00	0.00	0.00	0.00	0.0000
9	1.0	99.94	31980	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
2	5.0	95.51	25004	33.41	23.28	13.18	1.69	0.20	0.08	0.04	0.0000
3	5.0	95.93	30204	12.93	7.78	3.56	0.37	0.04	0.01	0.00	0.0000
4	5.0	96.79	30958	3.77	1.80	0.63	0.08	0.03	0.00	0.00	0.0000
5	5.0	97.38	31160	0.98	0.37	0.11	0.02	0.00	0.00	0.00	0.0000
6	5.0	97.94	31342	0.30	0.08	0.01	0.00	0.00	0.00	0.00	0.0000
9	5.0	98.45	31504	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
2	9.0	94.41	24674	29.85	21.12	12.10	1.56	0.16	0.06	0.02	0.0000
3	9.0	94.45	29769	10.88	6.50	3.03	0.36	0.02	0.01	0.00	0.0000
4	9.0	95.25	30467	2.98	1.51	0.54	0.06	0.01	0.00	0.00	0.0000
5	9.0	95.85	30671	0.67	0.26	0.07	0.01	0.01	0.01	0.00	0.0000
6	9.0	96.31	30819	0.16	0.03	0.01	0.00	0.00	0.00	0.00	0.0000
9	9.0	97.38	31163	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.0000

Table B.19. TRx values for degree $p = 2$, demand $d = 1$, mixed element latency functions, and fixed consumption number range.

n	$r_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	99.26	26118	47.49	35.81	22.93	6.71	1.83	0.69	0.35	0.0613
3	1.0	99.18	31259	24.74	16.19	9.25	3.87	1.44	0.68	0.36	0.0864
4	1.0	98.81	31598	10.85	6.01	3.03	1.33	0.53	0.28	0.15	0.0348
5	1.0	98.70	31584	5.10	2.39	1.19	0.54	0.30	0.20	0.14	0.0222
6	1.0	98.33	31466	2.67	1.13	0.50	0.24	0.12	0.08	0.04	0.0127
9	1.0	97.36	31154	0.84	0.35	0.22	0.13	0.08	0.04	0.02	0.0000
2	5.0	98.75	26007	32.63	23.51	13.67	2.83	0.45	0.04	0.02	0.0000
3	5.0	98.78	31126	13.29	8.23	4.37	1.28	0.31	0.05	0.01	0.0000
4	5.0	98.77	31583	4.35	2.23	0.98	0.28	0.08	0.02	0.01	0.0000
5	5.0	98.87	31637	1.19	0.42	0.17	0.07	0.02	0.01	0.00	0.0000
6	5.0	98.87	31638	0.42	0.14	0.04	0.02	0.00	0.00	0.00	0.0000
9	5.0	98.84	31630	0.06	0.01	0.01	0.01	0.00	0.00	0.00	0.0000
2	9.0	98.98	25936	27.69	19.95	11.57	2.35	0.45	0.05	0.02	0.0000
3	9.0	98.86	31148	10.77	6.50	3.35	0.87	0.22	0.06	0.01	0.0000
4	9.0	98.77	31587	2.96	1.43	0.54	0.15	0.03	0.00	0.00	0.0000
5	9.0	98.86	31634	0.81	0.31	0.10	0.03	0.01	0.00	0.00	0.0000
6	9.0	98.71	31588	0.25	0.07	0.03	0.01	0.01	0.01	0.00	0.0000
9	9.0	98.71	31586	0.03	0.02	0.01	0.01	0.01	0.00	0.00	0.0000

Table B.20. TRx values for degree $p = 2$, demand $d = 1$, mixed element latency functions, and fixed relevance number range.

n	$c_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	100.00	26349	41.97	30.90	18.40	3.31	0.42	0.00	0.00	0.0000
3	1.0	99.99	31503	19.97	12.21	5.85	1.05	0.16	0.02	0.00	0.0000
4	1.0	99.98	31974	7.82	3.81	1.30	0.23	0.04	0.00	0.00	0.0000
5	1.0	99.99	31996	2.86	1.00	0.29	0.04	0.02	0.00	0.00	0.0000
6	1.0	99.98	31995	1.11	0.33	0.06	0.02	0.00	0.00	0.00	0.0000
9	1.0	99.99	31996	0.13	0.03	0.00	0.00	0.00	0.00	0.00	0.0000
2	5.0	99.33	26299	31.09	22.38	13.05	2.57	0.37	0.03	0.00	0.0038
3	5.0	99.35	31319	13.15	8.05	4.04	1.02	0.25	0.03	0.01	0.0000
4	5.0	99.27	31753	4.11	1.96	0.74	0.17	0.06	0.03	0.00	0.0000
5	5.0	99.38	31800	1.14	0.41	0.16	0.06	0.03	0.01	0.00	0.0000
6	5.0	99.28	31771	0.44	0.11	0.03	0.01	0.00	0.00	0.00	0.0000
9	5.0	99.29	31774	0.04	0.01	0.00	0.00	0.00	0.00	0.00	0.0000
2	9.0	98.91	26112	28.24	20.27	11.57	2.38	0.41	0.03	0.00	0.0000
3	9.0	98.84	31139	11.32	7.04	3.58	0.99	0.27	0.05	0.01	0.0000
4	9.0	98.82	31609	3.10	1.47	0.65	0.16	0.03	0.01	0.01	0.0000
5	9.0	98.89	31645	0.76	0.33	0.10	0.03	0.01	0.00	0.00	0.0000
6	9.0	98.87	31637	0.23	0.06	0.03	0.01	0.00	0.00	0.00	0.0000
9	9.0	98.78	31609	0.03	0.02	0.01	0.00	0.00	0.00	0.00	0.0000

Table B.21. TRx values for degree $p = 2$, demand $d = 10$, single element latency functions, and fixed consumption number range.

n	$r_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	96.37	25102	49.95	39.02	26.76	7.71	2.82	1.96	1.34	0.1593
3	1.0	94.98	29893	24.34	15.96	8.87	3.25	1.13	0.63	0.42	0.0569
4	1.0	95.87	30661	11.96	7.15	3.81	1.76	0.69	0.40	0.30	0.0457
5	1.0	96.00	30721	6.73	3.51	1.73	0.81	0.37	0.23	0.16	0.0391
6	1.0	95.64	30605	4.34	2.01	0.93	0.45	0.25	0.17	0.10	0.0098
9	1.0	93.37	29878	2.06	0.79	0.36	0.19	0.14	0.06	0.04	0.0067
2	5.0	93.55	24300	34.00	23.75	13.29	1.67	0.07	0.01	0.01	0.0041
3	5.0	92.38	29087	13.12	7.95	3.64	0.40	0.04	0.01	0.01	0.0000
4	5.0	92.73	29660	3.86	1.77	0.62	0.06	0.01	0.01	0.00	0.0000
5	5.0	93.03	29770	0.98	0.32	0.08	0.01	0.00	0.00	0.00	0.0000
6	5.0	92.82	29702	0.26	0.04	0.01	0.00	0.00	0.00	0.00	0.0000
9	5.0	92.95	29743	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
2	9.0	93.54	24470	29.59	20.85	11.87	1.72	0.11	0.04	0.01	0.0082
3	9.0	92.16	28997	11.01	6.46	3.10	0.42	0.03	0.00	0.00	0.0000
4	9.0	92.20	29486	3.03	1.42	0.54	0.05	0.00	0.00	0.00	0.0000
5	9.0	91.83	29386	0.61	0.21	0.06	0.00	0.00	0.00	0.00	0.0000
6	9.0	91.83	29386	0.13	0.04	0.01	0.00	0.00	0.00	0.00	0.0000
9	9.0	91.58	29305	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000

Table B.22. TRx values for degree $p = 2$, demand $d = 10$, single element latency functions, and fixed relevance number range.

n	$c_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	100.00	26312	41.94	30.59	17.73	2.15	0.00	0.00	0.00	0.0000
3	1.0	99.27	31258	19.46	11.81	5.31	0.37	0.00	0.00	0.00	0.0000
4	1.0	99.30	31761	6.88	3.26	1.04	0.06	0.00	0.00	0.00	0.0000
5	1.0	99.26	31762	2.15	0.73	0.14	0.00	0.00	0.00	0.00	0.0000
6	1.0	99.32	31782	0.62	0.14	0.01	0.00	0.00	0.00	0.00	0.0000
9	1.0	99.34	31790	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
2	5.0	94.97	24753	33.53	23.61	13.16	1.72	0.08	0.02	0.00	0.0000
3	5.0	93.52	29431	12.61	7.58	3.39	0.41	0.01	0.01	0.00	0.0000
4	5.0	94.30	30157	3.67	1.68	0.59	0.05	0.01	0.00	0.00	0.0000
5	5.0	94.27	30166	0.89	0.27	0.07	0.02	0.00	0.00	0.00	0.0000
6	5.0	94.57	30263	0.24	0.05	0.00	0.00	0.00	0.00	0.00	0.0000
9	5.0	94.81	30338	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
2	9.0	93.69	24275	30.32	21.38	12.13	1.47	0.10	0.02	0.00	0.0000
3	9.0	91.79	28890	10.67	6.54	3.04	0.30	0.03	0.02	0.01	0.0000
4	9.0	91.87	29378	3.03	1.49	0.56	0.06	0.01	0.00	0.00	0.0000
5	9.0	91.67	29336	0.58	0.20	0.06	0.00	0.00	0.00	0.00	0.0000
6	9.0	91.51	29283	0.15	0.05	0.00	0.00	0.00	0.00	0.00	0.0000
9	9.0	91.54	29292	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000

Table B.23. TRx values for degree $p = 2$, demand $d = 10$, mixed element latency functions, and fixed consumption number range.

n	$r_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	97.81	25619	54.06	42.94	30.63	13.87	6.93	4.25	2.24	0.0351
3	1.0	98.03	30866	28.89	19.69	12.11	5.93	3.06	1.92	1.15	0.0065
4	1.0	98.13	31381	14.30	8.32	4.75	2.61	1.46	0.91	0.48	0.0064
5	1.0	97.80	31296	7.42	3.97	2.13	1.21	0.73	0.44	0.23	0.0032
6	1.0	97.36	31156	4.40	2.21	1.28	0.75	0.50	0.30	0.17	0.0032
9	1.0	95.79	30653	1.51	0.68	0.35	0.20	0.13	0.07	0.03	0.0033
2	5.0	94.77	24759	35.83	25.74	15.25	3.46	0.70	0.18	0.03	0.0000
3	5.0	94.17	29622	14.04	8.44	4.26	1.01	0.18	0.02	0.00	0.0000
4	5.0	94.57	30246	4.54	2.23	0.91	0.23	0.04	0.01	0.00	0.0000
5	5.0	95.45	30543	1.45	0.50	0.19	0.05	0.01	0.00	0.00	0.0000
6	5.0	95.88	30682	0.41	0.12	0.02	0.00	0.00	0.00	0.00	0.0000
9	5.0	97.11	31075	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
2	9.0	95.21	24905	30.97	22.17	12.98	2.69	0.53	0.12	0.02	0.0000
3	9.0	94.25	29695	11.80	7.41	3.74	0.94	0.19	0.05	0.00	0.0000
4	9.0	94.86	30338	3.35	1.79	0.78	0.21	0.03	0.01	0.00	0.0000
5	9.0	95.44	30542	0.82	0.33	0.12	0.02	0.00	0.00	0.00	0.0000
6	9.0	95.93	30696	0.30	0.08	0.02	0.00	0.00	0.00	0.00	0.0000
9	9.0	96.92	31014	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000

Table B.24. TRx values for degree $p = 2$, demand $d = 10$, mixed element latency functions, and fixed relevance number range.

n	$c_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	100.00	26253	42.49	31.44	18.58	3.26	0.50	0.01	0.00	0.0000
3	1.0	99.79	31432	20.43	12.76	6.17	1.48	0.23	0.02	0.00	0.0000
4	1.0	99.75	31898	8.10	4.09	1.61	0.42	0.08	0.02	0.00	0.0000
5	1.0	99.69	31902	3.06	1.09	0.37	0.08	0.01	0.00	0.00	0.0000
6	1.0	99.78	31928	1.20	0.29	0.08	0.02	0.01	0.00	0.00	0.0000
9	1.0	99.84	31950	0.19	0.03	0.00	0.00	0.00	0.00	0.00	0.0000
2	5.0	96.29	25326	34.46	24.63	14.43	3.12	0.62	0.11	0.02	0.0000
3	5.0	95.86	30166	14.07	8.75	4.33	1.06	0.22	0.05	0.00	0.0000
4	5.0	96.59	30895	4.25	2.11	0.93	0.20	0.04	0.00	0.00	0.0000
5	5.0	96.99	31037	1.25	0.48	0.15	0.03	0.01	0.01	0.00	0.0000
6	5.0	97.25	31119	0.39	0.12	0.03	0.01	0.01	0.00	0.00	0.0000
9	5.0	98.14	31405	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
2	9.0	95.26	25013	31.36	22.45	13.31	2.94	0.57	0.09	0.01	0.0000
3	9.0	94.31	29698	12.16	7.33	3.80	0.97	0.17	0.03	0.00	0.0000
4	9.0	94.86	30344	3.30	1.65	0.71	0.21	0.04	0.00	0.00	0.0000
5	9.0	95.48	30553	0.92	0.36	0.09	0.03	0.00	0.00	0.00	0.0000
6	9.0	95.83	30667	0.22	0.06	0.01	0.00	0.00	0.00	0.00	0.0000
9	9.0	96.92	31013	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.0000

Table B.25. TRx values for degree $p = 3$, demand $d = 1$, single element latency functions, and fixed consumption number range.

n	$r_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	71.86	8620	48.12	35.57	17.98	4.78	3.50	2.88	2.32	1.0673
3	1.0	89.54	14066	16.50	10.64	5.79	2.84	2.08	1.68	1.44	0.6043
4	1.0	94.41	15099	7.22	4.41	2.70	1.72	1.26	0.93	0.74	0.3908
5	1.0	93.31	14930	3.34	1.95	1.13	0.82	0.60	0.48	0.33	0.1741
6	1.0	92.04	14726	1.83	1.05	0.69	0.46	0.36	0.28	0.25	0.1494
9	1.0	90.64	14502	0.58	0.35	0.22	0.10	0.07	0.03	0.03	0.0207
2	5.0	88.84	11380	22.77	13.48	3.44	0.37	0.18	0.12	0.04	0.0088
3	5.0	90.41	14238	6.62	3.28	0.68	0.09	0.04	0.02	0.01	0.0000
4	5.0	92.69	14820	1.30	0.52	0.10	0.00	0.00	0.00	0.00	0.0000
5	5.0	94.03	15044	0.23	0.05	0.01	0.01	0.01	0.00	0.00	0.0000
6	5.0	95.38	15261	0.05	0.02	0.01	0.00	0.00	0.00	0.00	0.0000
9	5.0	96.81	15489	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
2	9.0	88.91	11434	18.55	11.12	2.64	0.24	0.14	0.04	0.01	0.0000
3	9.0	88.92	14005	5.81	2.93	0.59	0.04	0.03	0.01	0.00	0.0000
4	9.0	91.58	14637	1.18	0.46	0.08	0.01	0.01	0.00	0.00	0.0000
5	9.0	92.53	14804	0.18	0.06	0.00	0.00	0.00	0.00	0.00	0.0000
6	9.0	93.98	15037	0.03	0.02	0.00	0.00	0.00	0.00	0.00	0.0000
9	9.0	95.49	15279	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000

Table B.26. TRx values for degree $p = 3$, demand $d = 1$, single element latency functions, and fixed relevance number range.

n	$c_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	100.00	13163	27.53	16.80	3.63	0.00	0.00	0.00	0.00	0.0000
3	1.0	98.73	15562	9.72	4.67	0.63	0.00	0.00	0.00	0.00	0.0000
4	1.0	99.26	15868	2.47	0.88	0.05	0.00	0.00	0.00	0.00	0.0000
5	1.0	99.32	15890	0.60	0.13	0.00	0.00	0.00	0.00	0.00	0.0000
6	1.0	99.54	15927	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
9	1.0	99.81	15970	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
2	5.0	90.03	11588	21.25	12.69	2.99	0.23	0.14	0.06	0.02	0.0000
3	5.0	91.85	14477	6.70	3.26	0.59	0.08	0.03	0.01	0.00	0.0000
4	5.0	93.97	15028	1.35	0.49	0.07	0.01	0.01	0.00	0.00	0.0000
5	5.0	94.92	15187	0.20	0.05	0.01	0.01	0.00	0.00	0.00	0.0000
6	5.0	96.16	15386	0.05	0.01	0.01	0.00	0.00	0.00	0.00	0.0000
9	5.0	97.59	15615	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
2	9.0	88.83	11493	18.72	10.94	2.69	0.21	0.13	0.08	0.01	0.0000
3	9.0	89.33	14075	5.51	2.69	0.49	0.04	0.01	0.01	0.00	0.0000
4	9.0	91.37	14613	1.12	0.43	0.08	0.01	0.00	0.00	0.00	0.0000
5	9.0	92.53	14805	0.16	0.03	0.01	0.01	0.01	0.00	0.00	0.0000
6	9.0	93.69	14991	0.05	0.01	0.00	0.00	0.00	0.00	0.00	0.0000
9	9.0	95.73	15317	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000

Table B.27. TRx values for degree $p = 3$, demand $d = 1$, mixed element latency functions, and fixed consumption number range.

n	$r_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	97.38	12796	35.85	25.31	11.87	3.81	1.89	1.03	0.38	0.0156
3	1.0	98.01	15457	14.32	8.62	3.93	1.49	0.82	0.53	0.25	0.0712
4	1.0	97.73	15626	4.65	2.50	1.18	0.51	0.22	0.11	0.07	0.0192
5	1.0	97.12	15540	1.68	0.75	0.39	0.21	0.14	0.06	0.05	0.0064
6	1.0	96.31	15409	0.55	0.27	0.16	0.10	0.06	0.04	0.01	0.0065
9	1.0	94.23	15076	0.14	0.06	0.05	0.02	0.01	0.00	0.00	0.0000
2	5.0	95.38	12450	21.45	13.16	3.91	0.49	0.08	0.02	0.01	0.0000
3	5.0	96.33	15185	6.99	3.83	1.33	0.43	0.12	0.05	0.00	0.0000
4	5.0	97.09	15525	1.55	0.81	0.27	0.08	0.03	0.02	0.00	0.0000
5	5.0	97.44	15591	0.24	0.10	0.03	0.01	0.00	0.00	0.00	0.0000
6	5.0	97.86	15658	0.08	0.03	0.00	0.00	0.00	0.00	0.00	0.0000
9	5.0	98.09	15695	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
2	9.0	96.01	12638	18.09	11.09	3.46	0.59	0.17	0.03	0.00	0.0000
3	9.0	96.07	15140	5.67	3.22	1.26	0.33	0.10	0.01	0.01	0.0000
4	9.0	96.88	15493	1.45	0.64	0.23	0.07	0.03	0.01	0.00	0.0000
5	9.0	96.92	15507	0.19	0.04	0.02	0.00	0.00	0.00	0.00	0.0000
6	9.0	97.46	15593	0.03	0.01	0.01	0.00	0.00	0.00	0.00	0.0000
9	9.0	97.63	15621	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000

Table B.28. TRx values for degree $p = 3$, demand $d = 1$, mixed element latency functions, and fixed relevance number range.

n	$c_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	100.00	13135	26.80	16.51	4.24	0.21	0.03	0.00	0.00	0.0000
3	1.0	99.97	15734	9.95	5.17	1.34	0.18	0.05	0.01	0.00	0.0000
4	1.0	99.97	15984	2.83	1.08	0.20	0.04	0.01	0.00	0.00	0.0000
5	1.0	99.99	15998	0.69	0.26	0.07	0.03	0.00	0.00	0.00	0.0000
6	1.0	99.96	15994	0.13	0.03	0.01	0.00	0.00	0.00	0.00	0.0000
9	1.0	99.97	15995	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
2	5.0	97.38	12812	19.65	12.20	3.51	0.41	0.09	0.01	0.00	0.0000
3	5.0	97.78	15387	6.92	3.87	1.34	0.36	0.16	0.04	0.01	0.0000
4	5.0	97.81	15641	1.51	0.65	0.19	0.04	0.01	0.01	0.00	0.0000
5	5.0	98.35	15736	0.32	0.10	0.01	0.00	0.00	0.00	0.00	0.0000
6	5.0	98.29	15727	0.04	0.01	0.01	0.01	0.01	0.01	0.00	0.0000
9	5.0	98.35	15736	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
2	9.0	95.86	12593	18.91	11.74	3.78	0.50	0.17	0.02	0.00	0.0000
3	9.0	96.38	15163	5.74	3.46	1.35	0.34	0.11	0.03	0.00	0.0000
4	9.0	96.58	15444	1.21	0.56	0.27	0.04	0.01	0.01	0.00	0.0000
5	9.0	97.16	15545	0.23	0.08	0.03	0.00	0.00	0.00	0.00	0.0000
6	9.0	97.26	15561	0.03	0.01	0.00	0.00	0.00	0.00	0.00	0.0000
9	9.0	97.88	15660	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.0000

Table B.29. TRx values for degree $p = 3$, demand $d = 10$, single element latency functions, and fixed consumption number range.

n	$r_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	87.41	11163	40.65	30.28	15.30	4.64	3.20	2.46	2.01	0.6181
3	1.0	88.11	13859	16.00	10.17	5.06	2.02	1.16	0.85	0.63	0.2381
4	1.0	90.64	14489	6.44	3.73	2.18	1.24	0.81	0.64	0.50	0.1242
5	1.0	91.27	14603	3.23	1.86	1.06	0.68	0.52	0.40	0.27	0.0685
6	1.0	89.09	14255	1.92	1.10	0.70	0.48	0.32	0.20	0.14	0.0351
9	1.0	87.09	13934	0.69	0.42	0.32	0.17	0.12	0.09	0.06	0.0072
2	5.0	86.58	11125	22.52	13.56	3.02	0.11	0.04	0.03	0.01	0.0000
3	5.0	86.43	13579	6.61	3.57	0.68	0.02	0.01	0.01	0.00	0.0000
4	5.0	87.25	13952	1.52	0.59	0.10	0.02	0.00	0.00	0.00	0.0000
5	5.0	86.91	13906	0.27	0.06	0.00	0.00	0.00	0.00	0.00	0.0000
6	5.0	87.61	14017	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
9	5.0	88.46	14154	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
2	9.0	88.08	11229	19.57	11.63	2.48	0.20	0.04	0.02	0.01	0.0000
3	9.0	85.61	13433	5.69	2.57	0.54	0.04	0.02	0.00	0.00	0.0000
4	9.0	84.79	13557	1.05	0.46	0.04	0.00	0.00	0.00	0.00	0.0000
5	9.0	85.05	13608	0.12	0.04	0.01	0.00	0.00	0.00	0.00	0.0000
6	9.0	85.41	13665	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.0000
9	9.0	85.77	13723	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000

Table B.30. TRx values for degree $p = 3$, demand $d = 10$, single element latency functions, and fixed relevance number range.

n	$c_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	100.00	13202	26.96	16.53	3.39	0.00	0.00	0.00	0.00	0.0000
3	1.0	98.39	15486	9.82	4.67	0.61	0.00	0.00	0.00	0.00	0.0000
4	1.0	98.00	15676	2.39	0.82	0.11	0.00	0.00	0.00	0.00	0.0000
5	1.0	97.55	15608	0.49	0.15	0.01	0.00	0.00	0.00	0.00	0.0000
6	1.0	97.44	15590	0.12	0.02	0.00	0.00	0.00	0.00	0.00	0.0000
9	1.0	97.80	15648	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
2	5.0	88.78	11397	20.65	11.80	2.54	0.18	0.11	0.04	0.00	0.0000
3	5.0	87.85	13809	6.83	3.55	0.70	0.02	0.00	0.00	0.00	0.0000
4	5.0	88.09	14088	1.40	0.63	0.06	0.00	0.00	0.00	0.00	0.0000
5	5.0	89.04	14247	0.27	0.12	0.02	0.00	0.00	0.00	0.00	0.0000
6	5.0	89.66	14345	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
9	5.0	90.96	14554	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
2	9.0	87.74	11210	19.01	11.38	2.41	0.20	0.11	0.04	0.00	0.0000
3	9.0	85.58	13459	5.71	2.82	0.51	0.01	0.01	0.01	0.00	0.0000
4	9.0	84.91	13578	1.13	0.47	0.05	0.01	0.01	0.00	0.00	0.0000
5	9.0	85.21	13633	0.15	0.04	0.00	0.00	0.00	0.00	0.00	0.0000
6	9.0	85.52	13684	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
9	9.0	85.62	13700	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000

Table B.31. TRx values for degree $p = 3$, demand $d = 10$, mixed element latency functions, and fixed consumption number range.

n	$r_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	97.31	12793	42.83	33.00	20.07	10.62	7.65	5.46	3.66	0.2032
3	1.0	97.06	15281	18.79	12.48	7.35	4.30	3.24	2.54	1.94	0.2094
4	1.0	96.16	15377	8.79	5.54	3.49	2.52	1.82	1.38	0.98	0.1301
5	1.0	95.29	15247	3.96	2.59	1.76	1.30	1.06	0.76	0.54	0.0853
6	1.0	93.41	14946	2.13	1.34	0.84	0.63	0.43	0.29	0.19	0.0201
9	1.0	90.97	14556	0.80	0.50	0.34	0.23	0.18	0.10	0.05	0.0069
2	5.0	89.02	11423	23.16	13.72	3.81	0.56	0.19	0.05	0.02	0.0000
3	5.0	87.88	13789	7.46	3.92	0.96	0.16	0.05	0.03	0.01	0.0000
4	5.0	88.96	14224	1.43	0.58	0.16	0.04	0.01	0.01	0.00	0.0000
5	5.0	90.38	14460	0.18	0.04	0.02	0.01	0.00	0.00	0.00	0.0000
6	5.0	91.89	14702	0.04	0.01	0.01	0.01	0.00	0.00	0.00	0.0000
9	5.0	94.16	15065	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
2	9.0	89.38	11548	19.88	12.32	3.06	0.43	0.20	0.09	0.04	0.0000
3	9.0	87.87	13807	5.87	2.80	0.73	0.07	0.04	0.01	0.01	0.0000
4	9.0	88.92	14218	1.09	0.50	0.12	0.02	0.00	0.00	0.00	0.0000
5	9.0	89.85	14376	0.19	0.08	0.04	0.00	0.00	0.00	0.00	0.0000
6	9.0	90.61	14498	0.06	0.02	0.00	0.00	0.00	0.00	0.00	0.0000
9	9.0	93.39	14943	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000

Table B.32. TRx values for degree $p = 3$, demand $d = 10$, mixed element latency functions, and fixed relevance number range.

n	$c_{\max}$	%OK	#($\gamma > 1$)	%TR30	%TR40	%TR50	%TR60	%TR70	%TR80	%TR90	%TR100
2	1.0	100.00	13197	27.49	16.38	4.20	0.22	0.02	0.00	0.00	0.0000
3	1.0	99.41	15644	9.74	4.99	1.20	0.08	0.01	0.00	0.00	0.0000
4	1.0	99.06	15841	2.66	1.16	0.21	0.02	0.00	0.00	0.00	0.0000
5	1.0	98.76	15802	0.68	0.18	0.03	0.00	0.00	0.00	0.00	0.0000
6	1.0	98.78	15804	0.18	0.03	0.01	0.00	0.00	0.00	0.00	0.0000
9	1.0	99.34	15894	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
2	5.0	91.17	11797	22.81	14.04	3.93	0.47	0.19	0.05	0.00	0.0000
3	5.0	90.49	14254	6.70	3.39	0.86	0.15	0.03	0.01	0.01	0.0000
4	5.0	91.67	14654	1.34	0.57	0.12	0.02	0.00	0.00	0.00	0.0000
5	5.0	92.63	14821	0.31	0.09	0.01	0.01	0.00	0.00	0.00	0.0000
6	5.0	93.40	14944	0.07	0.02	0.00	0.00	0.00	0.00	0.00	0.0000
9	5.0	95.44	15270	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
2	9.0	89.66	11464	20.80	12.76	3.89	0.56	0.19	0.09	0.02	0.0000
3	9.0	88.22	13882	5.75	2.99	0.71	0.07	0.03	0.01	0.00	0.0000
4	9.0	89.14	14256	1.23	0.56	0.13	0.01	0.01	0.00	0.00	0.0000
5	9.0	89.99	14399	0.20	0.05	0.00	0.00	0.00	0.00	0.00	0.0000
6	9.0	90.87	14539	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
9	9.0	93.57	14971	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.0000

B.3 Comparison with Perakis' Bound

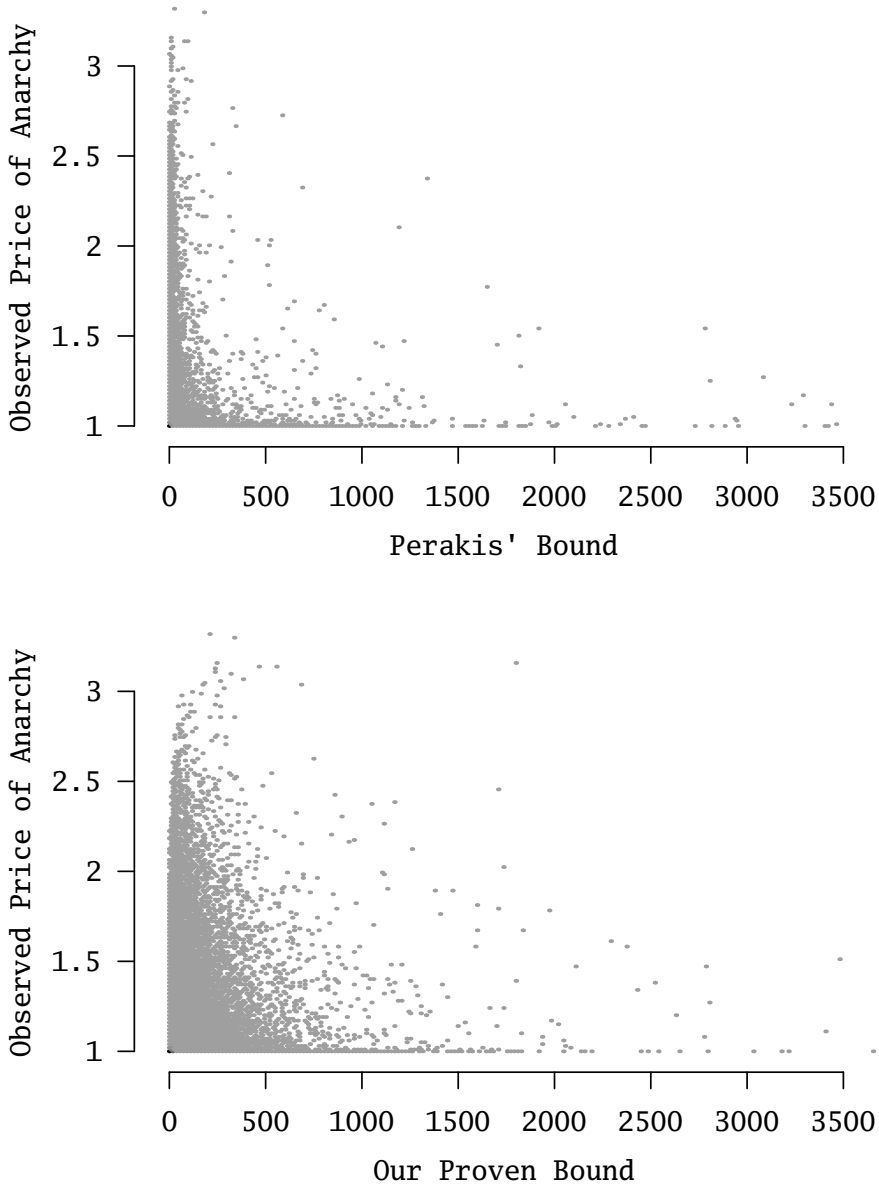


Figure B.8. Hexagonal binning plots with the observed price of anarchy over Perakis' bound and our proven bound of Thm. 2.12, respectively. The data set comprises 926,254 instances. The horizontal scale is trimmed to the largest value of our bound, resulting in 10 values of Perakis' bound being omitted (in the upper plot). Those values of Perakis' bound range up to 9,007.

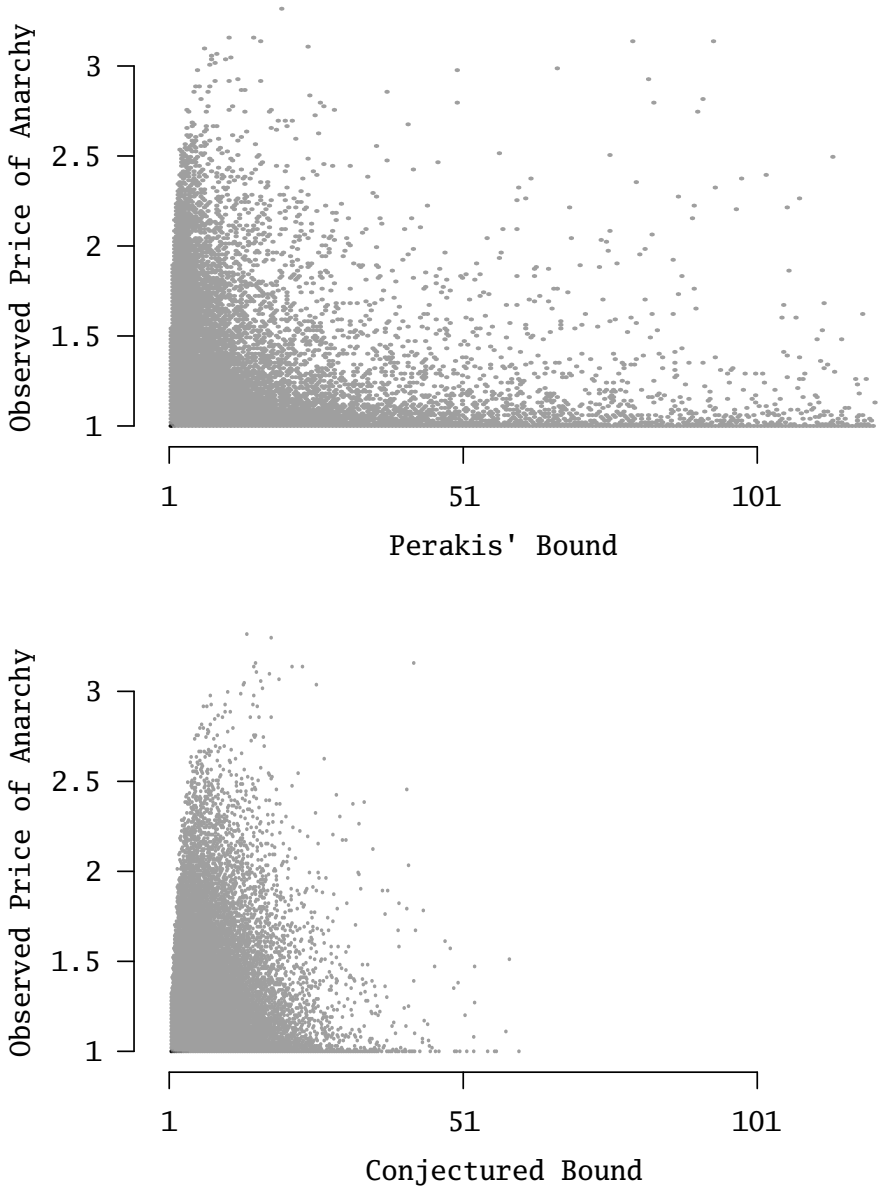


Figure B.9. Hexagonal binning plots with the observed price of anarchy over Perakis' bound and our conjectured bound, respectively. We use the same data set as for the previous plot. The horizontal scale is trimmed to two times the maximum of our conjectured bound, resulting in 1,395 instances being omitted (in the upper plot).

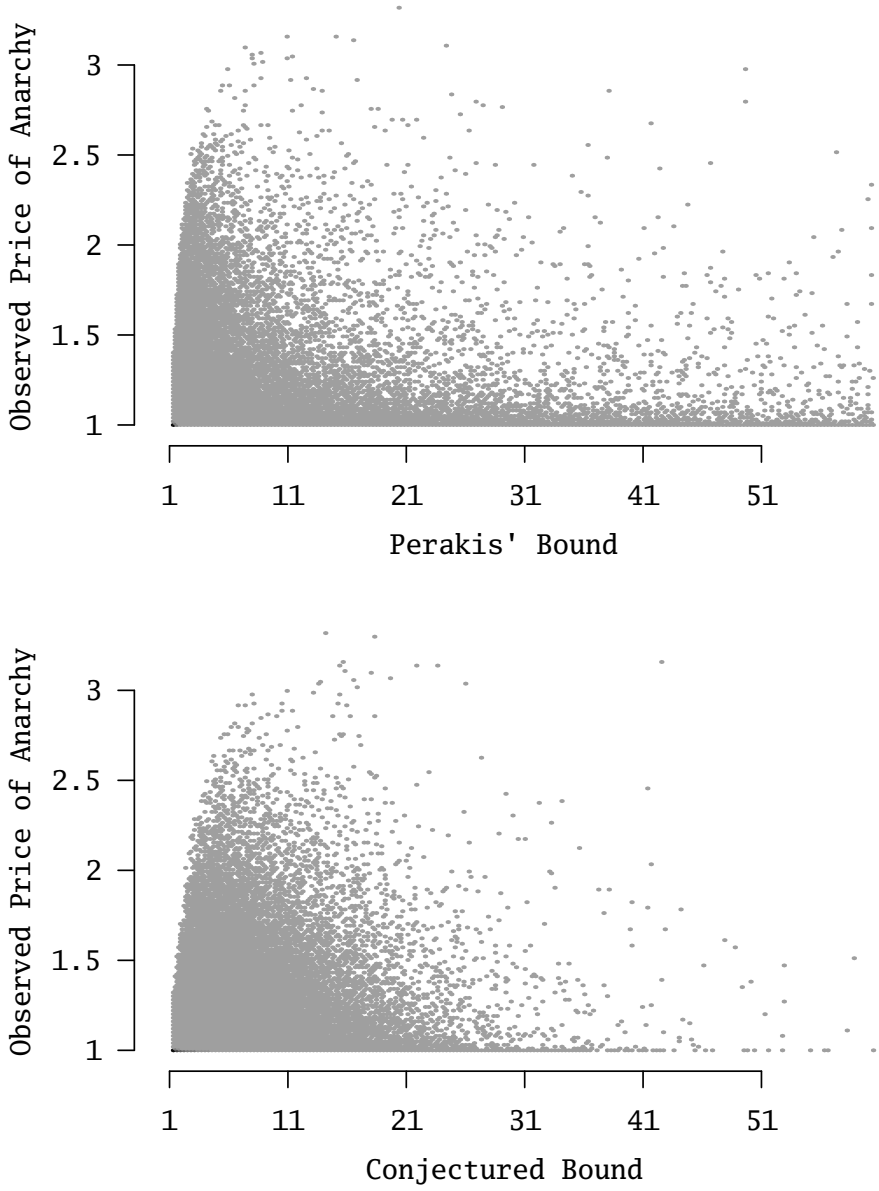


Figure B.10. Observed price of anarchy over Perakis' bound and our conjectured bound, respectively. These are magnifications of the left parts of the plots from the previous page; we trimmed the horizontal scale to the maximum value of our conjectured bound, resulting in 2,903 instances being omitted (in the upper plot).

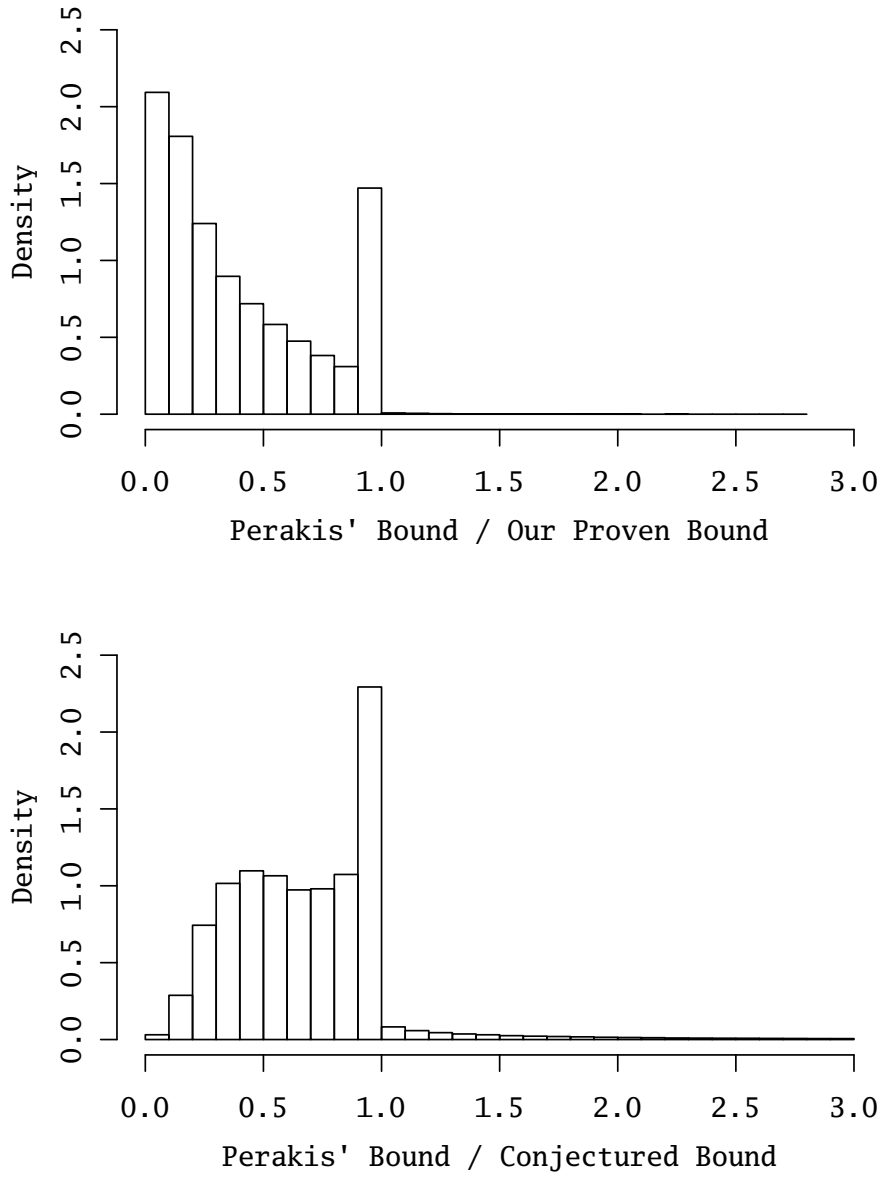


Figure B.11. Histograms for the ratio between Perakis' bound and our bounds: in the first we compare to our proven bound and in the second we compare to our conjectured bound. The underlying data set is the same as for the previous plots. However, only those instances were chosen to contribute to the histograms for which both ratios – namely Perakis' bound divided by our proven bound and Perakis' bound divided by our conjectured bound – are at most 3. This results in 912,152 instances contributing to each of the histograms and 14,102 being omitted each.

Bibliography

[1] Michele Aghassi, Dimitris Bertsimas, and Georgia Perakis. Solving asymmetric variational inequalities via convex optimization. *Operations Research Letters*, 34(5):481–490, 2006. URL http://dx.doi.org/10.1016/j.orl.2005.09.006.

[2] Susanne Albers, Stefan Eilts, Eyal Even-Dar, Yishay Mansour, and Liam Roditty. On Nash equilibria for a network creation game. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms, Miami, Florida, USA, January 2006 (SODA 2006)*, pages 89–98, 2006. URL http://dx.doi.org/10.1145/1109557.1109568.

[3] Réka Albert, Hawoong Jeong, and Albert-László Barabási. Error and attack tolerance of complex networks. *Nature*, 406:378–382, 2000. URL http://dx.doi.org/10.1038/35019019.

[4] Elliot Anshelevich, Anirban Dasgupta, Jon Kleinberg, Éva Tardos, Tom Wexler, and Tim Roughgarden. The price of stability for network design with fair cost allocation. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, Rome, Italy, October 2004 (FOCS 2004)*, pages 295 – 304, 2004. URL http://dx.doi.org/10.1109/FOCS.2004.68.

[5] Elliot Anshelevich, Anirban Dasgupta, Éva Tardos, and Tom Wexler. Near-optimal network design with selfish agents. *Theory of Computing*, 4:77–109, 2008. URL http://dx.doi.org/10.4086/toc.2008.v004a004. Conference version at STOC 2003.

[6] Baruch Awerbuch, Yossi Azar, and Amir Epstein. The price of

routing unsplittable flow. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 2005 (STOC 2005)*, pages 57–66, 2005. URL http://dx.doi.org/10.1145/1060590.1060599.

[7] Venkatesh Bala and Sanjeev Goyal. A noncooperative model of network formation. *Econometrica*, 68(5):1181–1229, 2000. URL http://dx.doi.org/10.1111/1468-0262.00155.

[8] Venkatesh Bala and Sanjeev Goyal. A strategic analysis of network reliability. *Review of Economic Design*, 5(3):205–228, 2000. URL http://dx.doi.org/10.1007/s100580000019.

[9] Andreas Baltz, Sandro Esquivel, Lasse Kliemann, and Anand Srivastav. The price of anarchy in selfish multicast routing. In *Revised Papers of the 3rd Workshop on Combinatorial and Algorithmic Aspects of Networking, Chester, UK, June 2006 (CAAN 2006)*, pages 5–18, 2006. URL http://dx.doi.org/10.1007/11922377_2.

[10] Nadine Baumann and Sebastian Stiller. The price of anarchy of a network creation game with exponential payoff. In *Proceedings of the 1st Annual ACM-SIAM Symposium on Algorithmic Game Theory, Paderborn, Germany, April and May 2008 (SAGT 2008)*, pages 218–229, 2008. URL http://dx.doi.org/10.1007/978-3-540-79309-0_20.

[11] Martin Beckmann, Charles B. McGuire, and Christopher B. Winsten. *Studies in the Economics of Transportation*. Yale University Press, 1956. URL http://www.rand.org/publications/RM/RM1488.pdf.

[12] Aharon Ben-Tal and Arkadi Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. SIAM, 2001. ISBN 0-89871-491-5.

[13] Francis Bloch and Matthew O. Jackson. Definitions of equilibrium in network formation games. *International Journal of Game Theory*, 34(3):305–318, 2006. URL http://dx.doi.org/10.1007/s00182-006-0022-9.

[14] Francis Bloch and Matthew O. Jackson. The formation of networks with transfers among players. *Journal of Economic Theory*, 133(1): 83 – 110, 2007. URL http://dx.doi.org/10.1016/j.jet.2005.10.003.

[15] Matthias Blonski. Anonymous games with binary actions. *Games and Economic Behavior*, 28(2):171–180, August 1999. URL http://dx.doi.org/10.1006/game.1998.0699.

[16] Thomas Boulogne and Eitan Altman. Competitive routing in multicast communications. *Networks*, 46(1):22–35, 2005. URL http://dx.doi.org/10.1002/net.20071.

[17] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004. ISBN 978-0521833783. URL http://www.stanford.edu/~boyd/cvxbook/.

[18] Dietrich Braess. Über ein Paradoxon aus der Verkehrsplanung. *Unternehmensforschung*, 12:258–268, 1968. URL http://homepage.ruhr-uni-bochum.de/Dietrich.Braess/paradox.pdf.

[19] Dietrich Braess, Anna Nagurney, and Tina Wakolbinger. On a paradox of traffic planning. *Transportation Science*, 39:446–450, 2005. URL http://homepage.rub.de/Dietrich.Braess/Paradox-BNW.pdf. Translation from the original German.

[20] Ulrik Brandes, Martin Hoefer, and Bobo Nick. Network creation games with disconnected equilibria. In *Proceedings of the 4th International Workshop on Internet and Network Economics, Shanghai, China, December 2008 (WINE 2008)*, pages 394–401, 2008. URL http://dx.doi.org/10.1007/978-3-540-92185-1_45. Available also at http://arxiv.org/abs/0805.4323.

[21] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the Web. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 33:309 – 320, 2000. URL http://dx.doi.

org/10.1016/S1389-1286(00)00083-9. Available also at http://www9.org/w9cdrom/160/160.html.

[22] Duncan S. Callaway, M.E.J. Newman, Steven H. Strogatz, and Duncan J. Watts. Network robustness and fragility: Percolation on random graphs. *Physical Review Letters*, 85(25):5468–5471, 2000. URL http://dx.doi.org/10.1103/PhysRevLett.85.5468.

[23] Antoni Calvó-Armengol and Rahmi İlkiliç. Pairwise stability and Nash equilibria in network formation. Technical Report 34.2005, Fondazione Eni Enrico Mattei, 2005. URL http://ssrn.com/abstract=677995.

[24] Dan Carr, Nicholas Lewin-Koh, and Martin Mächler. Hexagonal binning routines. URL http://www.bioconductor.org/packages/bioc/html/hexbin.html.

[25] Chi Kin Chau and Kwang Mong Sim. The price of anarchy for nonatomic congestion games with symmetric cost maps and elastic demands. *Operations Research Letters*, 31(5):327–334, 2003. URL http://dx.doi.org/10.1016/S0167-6377(03)00030-0.

[26] Nicolas Christin and John Chuang. On the cost of participating in a peer-to-peer network. In *Revised Selected Papers of the 3rd International Workshop on Peer-to-Peer Systems, La Jolla, CA, USA, February 2004 (IPTPS 2004)*, pages 22–32, 2005. URL http://dx.doi.org/10.1007/b104020. Available also at http://www.iptps.org/papers-2004/christin-cost.pdf.

[27] Nicolas Christin, Jens Grossklags, and John Chuang. Near rationality and competitive equilibria in networked systems. In *Proceedings of the ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems, Portland, Oregon, USA, August and September 2004 (SIGCOMM 2004)*, 2004. URL http://dx.doi.org/10.1145/1016527.1016536. Available also at http://conferences.sigcomm.org/sigcomm/2004/workshop_papers/pin34-christin1.pdf.

[28] George Christodoulou and Elias Koutsoupias. The price of anarchy of finite congestion games. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 2005 (STOC 2005)*, pages 67–73, 2005. URL http://doi.acm.org/10.1145/1060590.1060600.

[29] Byung-Gon Chun, Rodrigo Fonseca, Ion Stoica, and John Kubiatowicz. Characterizing selfishly constructed overlay routing networks. In *Proceedings of the 23rd IEEE Conference on Computer Communications, Hong Kong, China, March 2004 (INFOCOM 2004)*, 2004. URL http://www.ieee-infocom.org/2004/Papers/28_4.PDF.

[30] Roberto Cominetti, José R. Correa, and Nicolás E. Stier Moses. The impact of oligopolistic competition in networks. *Operations Research*, 57(6):1421–1437, 2009. doi: 10.1287/opre.1080.0653. URL http://www.dii.uchile.cl/~jcorrea/papers/CCS2009.pdf. Conference version at ICALP 2006.

[31] Andrew R. Conn, Nicholas I.M. Gould, and Philippe L. Toint. *Lancelot: A FORTRAN Package for Large-Scale Nonlinear Optimization (Release A)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1992. ISBN 0-387-55470-X.

[32] Jacomo Corbo and David Parkes. The price of selfish behavior in bilateral network formation. In *Proceedings of the 24th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, Las Vegas, Nevada, USA, July 2005 (PODC 2005)*, pages 99–107, 2005. URL http://dx.doi.org/10.1145/1073814.1073833. The conference version contains some errors. A revised version is in preparation.

[33] José R. Correa, Andreas S. Schulz, and Nicolás E. Stier Moses. Selfish routing in capacitated networks. *Mathematics of Operations Research*, 29(4):961–976, 2004. URL http://dx.doi.org/10.1287/moor.1040.0098.

[34] José R. Correa, Andreas S. Schulz, and Nicolás E. Stier Moses. A geometric approach to the price of anarchy in nonatomic con-

gestion games. *Games and Economic Behavior*, 64(2):457–469, 2008. URL http://dx.doi.org/10.1016/j.geb.2008.01.001.

[35] Artur Czumaj. Selfish routing on the Internet. In Joseph Y-T. Leung, editor, *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, chapter 42. CRC Press, Boca Raton, FL, USA, 2004.

[36] Artur Czumaj and Berthold Vöcking. Tight bounds for worst-case equilibria. *ACM Transactions on Algorithms*, 3(1):1–17, 2007. URL http://dx.doi.org/10.1145/1219944.1219949. Conference version at SODA 2002.

[37] Stella C. Dafermos and Frederick T. Sparrow. The traffic assignment problem for a general network. *Journal of Research of the National Bureau of Standards*, 73B(2):91–118, 1969.

[38] Erik D. Demaine, Mohammad Taghi Hajiaghayi, Hamid Mahini, and Morteza Zadimoghaddam. The price of anarchy in network creation games. In *Proceedings of the 26th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, Portland, Oregon, USA, August 2007 (PODC 2007)*, pages 292–298, 2007. URL http://doi.acm.org/10.1145/1281100.1281142.

[39] Erik D. Demaine, Mohammad Taghi Hajiaghayi, Hamid Mahini, and Morteza Zadimoghaddam. The price of anarchy in cooperative network creation games. In *Proceedings of the 26th Annual Symposium on Theoretical Aspects of Computer Science, Freiburg, Germany, February 2009 (STACS 2009)*, pages 301–312, 2009. URL http://drops.dagstuhl.de/opus/volltexte/2009/1839/.

[40] Reinhard Diestel. *Graph Theory*. Springer-Verlag New York, Inc., third edition, 2005. ISBN 3-540-26182-6. URL http://diestel-graph-theory.com/index.html.

[41] Alex Fabrikant, Ankur Luthra, Elitza Maneva, Christos H. Papadimitriou, and Scott Shenker. On a network creation game. In *Proceedings of the 22nd Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, Boston, Massachusetts,*

USA, July 2003 (PODC 2003), pages 347–351, 2003. URL http://dx.doi.org/10.1145/872035.872088.

[42] Dimitris Fotakis, Spyros Kontogiannis, and Paul Spirakis. Selfish unsplittable flows. *Theoretical Computer Science*, 348:226 – 239, 2005. URL http://dx.doi.org/10.1016/j.tcs.2005.09.024. Conference version at ICALP 2004.

[43] Eric J. Friedman. Dynamics and rationality in ordered externality games. *Games and Economic Behavior*, 16(1):65–76, 1996. URL http://dx.doi.org/10.1006/game.1996.0074.

[44] Mark Galassi, Jim Davies, James Theiler, Brian Gough, Gerard Jungman, Michael Booth, and Fabrice Rossi. *GNU Scientific Library: Reference Manual*. Network Theory Ltd., Bristol, UK, second edition, 2002. ISBN 0-9541617-3-4. URL http://www.gnu.org/software/gsl.

[45] Yair Halevi and Yishay Mansour. A network creation game with nonuniform interests. In *Proceedings of the 3rd International Workshop on Internet and Network Economics, San Diego, CA, USA, December 2007 (WINE 2007)*, pages 287–292, 2007. URL http://dx.doi.org/10.1007/978-3-540-77105-0_28.

[46] Hans Haller and Sudipta Sarangi. Nash networks with heterogeneous links. *Mathematical Social Sciences*, 50(2):181–201, 2005. URL http://dx.doi.org/10.1016/j.mathsocsci.2005.02.003. See also the related working paper [82] from 2003.

[47] Deren Han, Hong Kam Lo, and Hai Yang. On the price of anarchy for non-atomic congestion games under asymmetric cost maps and elastic demands. *Computers and Mathematics with Applications*, 56:2737–2743, 2008. URL http://dx.doi.org/10.1016/j.camwa.2008.05.038.

[48] Petter Holme, Beom Jun Kim, Chang No Yoon, and Seung Kee Han. Attack vulnerability of complex networks. *Physical Review E*, 65(5):14 pages, 2002. URL http://link.aps.org/doi/10.1103/PhysRevE.65.056109.

[49] Matthew O. Jackson. A survey of models of network formation: Stability and efficiency. In Gabrielle Demange and Myrna Wooders, editors, *Group Formation in Economics; Networks, Clubs and Coalitions*, chapter 1. Cambridge University Press, 2004. ISBN 978-0521842716. URL http://www.stanford.edu/~jacksonm/netsurv.pdf.

[50] Matthew O. Jackson and Asher Wolinsky. A strategic model of social and economic networks. *Journal of Economic Theory*, 71(1):44–74, 1996. URL http://dx.doi.org/10.1006/jeth.1996.0108.

[51] Ramesh Johari, Shie Mannor, and John N. Tsitsiklis. A contract-based model for directed network formation. *Games and Economic Behavior*, 56(2):201–224, 2006. URL http://dx.doi.org/10.1016/j.geb.2005.08.010.

[52] David Kinderlehrer and Guido Stampacchia. *An Introduction to Variational Inequalities and Applications*. Academic Press, New York, NY, 1980. ISBN 978-0898714661.

[53] Lasse Kliemann and Anand Srivastav. Models of non-atomic congestion games – from unicast to multicast routing. In Jürgen Lerner, Dorothea Wagner, and Katharina Zweig, editors, *Algorithmics of Large and Complex Networks*, volume 5515 of *Lecture Notes in Computer Science*, pages 292–318. Springer-Verlag Berlin Heidelberg, 2009. URL http://dx.doi.org/10.1007/978-3-642-02094-0_14.

[54] Elias Koutsoupias and Christos H. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science, Trier, Germany, March 1999 (STACS 1999)*, pages 404–413, 1999. URL http://dx.doi.org/10.1007/3-540-49116-3_38.

[55] Mikhail K. Kozlov, Sergei P. Tarasov, and Leonid G. Khachiyan. Polynomial solvability of convex quadratic programming. *Soviet Mathematics Doklady*, 20:1108–1111, 1979.

[56] Nikolaos Laoutaris, Laura J. Poplawski, Rajmohan Rajaraman, Ravi Sundaram, and Shang-Hua Teng. Bounded budget connection (BBC) games or how to make friends and influence people, on a budget. In *Proceedings of the 27th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, Toronto, Canada, August 2008 (PODC 2008)*, pages 165–174, 2008. URL http://doi.acm.org/10.1145/1400751.1400774. Available also at http://arxiv.org/abs/0806.1727.

[57] Henry Lin. On the price of anarchy of a network creation game. Unpublished manuscript, December 2003.

[58] Thomas Lücking, Marios Mavronicolas, Burkhard Monien, Manuel Rode, Paul G. Spirakis, and Imrich Vrto. Which is the worst-case Nash equilibrium? In *Proceedings of the 28th International Symposium on Mathematical Foundations of Computer Science, Bratislava, Slovakia, August 2003 (MFCS 2003)*, pages 551–561, 2003. URL http://dx.doi.org/10.1007/b11836.

[59] Thomas Lücking, Marios Mavronicolas, Burkhard Monien, and Manuel Rode. A new model for selfish routing. *Theoretical Computer Science*, 406(3):187–206, 2008. URL http://dx.doi.org/10.1016/j.tcs.2008.06.045. Conference version at STACS 2004.

[60] Andreu Mas-Colell. On a theorem of schmeidler. *Journal of Mathematical Economics*, 13(3):201–206, 1984. URL http://dx.doi.org/10.1016/0304-4068(84)90029-6.

[61] Igal Milchtaich. Congestion models of competition. *The American Naturalist*, 147:760–783, 1996. URL http://www.jstor.org/pss/2463089. Available also at http://faculty.biu.ac.il/~milchti/papers/competition.pdf.

[62] Igal Milchtaich. Generic uniqueness of equilibrium in large crowding games. *Mathematics of Operations Research*, 25(3):349–364, 2000. URL http://dx.doi.org/10.1287/moor.25.3.349.12220. Available also at http://faculty.biu.ac.il/~milchti/papers/nonatom.pdf.

[63] Igal Milchtaich. Social optimality and cooperation in nonatomic congestion games. *Journal of Economic Theory*, 114(1):56–87, 2004. URL http://dx.doi.org/10.1016/S0022-0531(03)00106-6. Available also at http://faculty.biu.ac.il/~milchti/papers/value.pdf.

[64] Igal Milchtaich. Topological conditions for uniqueness of equilibrium in networks. *Mathematics of Operations Research*, 30(1): 225–244, 2005. URL http://dx.doi.org/10.1287/moor.1040.0122. Available also at http://faculty.biu.ac.il/~milchti/papers/uniqueness.pdf.

[65] Igal Milchtaich. Network topology and the efficiency of equilibrium. *Games and Economic Behavior*, 57(2):321–346, 2006. URL http://dx.doi.org/10.1016/j.geb.2005.09.005. Available also at http://faculty.biu.ac.il/~milchti/papers/network.pdf.

[66] Thomas Moscibroda, Stefan Schmid, and Roger Wattenhofer. On the topologies formed by selfish peers. In *Proceedings of the 25th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, Denver, Colorado, USA, July 2006 (PODC 2006)*, pages 133–142, 2006. URL http://dx.doi.org/10.1145/1146381.1146403.

[67] Roger B. Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, Cambridge, Massachusetts and London, England, fifth printing edition, 2002. First printing in 1991.

[68] Anna Nagurney. *Network Economics: A Variational Inequality Approach*. Springer, Berlin, 1999. ISBN 978-0792383505.

[69] John F. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):289–295, 1951. URL http://www.jstor.org/stable/1969529.

[70] Yuri Nesterov and Arkadi Nemirovski. *Interior-Point Polynomial Algorithms in Convex Programming*. Studies in Applied Mathematics. SIAM, third printing edition, 2001. ISBN 978-0-898715-15-6.

[71] Christos H. Papadimitriou. Algorithms, games, and the Internet. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, Crete, Greece, July 2001 (STOC 2001)*, pages 749–753, 2001. URL http://dx.doi.org/10.1145/380752.380883. Extended abstract at ICALP 2001.

[72] Georgia Perakis. The price of anarchy when costs are non-separable and asymmetric. *Mathematics of Operations Research*, 32 (3):614–628, 2007. URL http://dx.doi.org/10.1287/moor.1070.0258. Conference version at IPCO 2004.

[73] Arthur C. Pigou. *The Economics of Welfare*. Macmillan, London, 1920.

[74] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. URL http://www.R-project.org. ISBN 3-900051-07-0.

[75] Kali P. Rath. A direct proof of the existence of pure strategy equilibria in games with a continuum of players. *Economic Theory*, 2(3):427–433, 1992. URL http://dx.doi.org/10.1007/BF01211424.

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

[77] Tim Roughgarden. *Selfish Routing*. PhD thesis, Cornell University, Ithaca, NY, USA, 2002. URL http://theory.stanford.edu/~tim/papers/thesis.pdf.

[78] Tim Roughgarden. *Selfish Routing and the Price of Anarchy*. MIT Press, 2005. ISBN 978-0-262-18243-0.

[79] Tim Roughgarden. Selfish routing with atomic players. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, Vancouver, British Columbia, Canada, January 2005 (SODA 2005)*, pages 1184–1185, 2005. URL http://doi.acm.org/10.1145/1070432.1070600. See also [30] for errata.

[80] Tim Roughgarden and Éva Tardos. Bounding the inefficiency of equilibria in nonatomic congestion games. *Games and Economic Behavior*, 47(2):389–403, 2004. URL http://dx.doi.org/10.1016/j.geb.2003.06.004.

[81] Jared Saia, Amos Fiat, Steve Gribble, Anna R. Karlin, and Stefan Saroiu. Dynamically fault-tolerant content addressable networks. In *Revised Papers of the 1st International Workshop on Peer-to-Peer Systems, Cambridge, MA, USA, March 2002 (IPTPS 2002)*, pages 270 – 279, 2002. URL http://dx.doi.org/10.1007/3-540-45748-8_26.

[82] Sudipta Sarangi and Hans Haller. Nash networks with heterogeneous agents. Departmental Working Papers 2003-06, Department of Economics, Louisiana State University, 2003. URL http://ideas.repec.org/p/lsu/lsuwpp/2003-06.html.

[83] David Schmeidler. Equilibrium points of nonatomic games. *Journal of Statistical Physics*, 7(4):295–300, April 1973. URL http://www.tau.ac.il/~schmeid/PDF/eq_of_nonatomic_games_1972.pdf.

[84] M.J. Smith. The existence, uniqueness and stability of traffic equilibria. *Transportation Research Part B: Methodological*, 13(4):295–304, 1979. URL http://dx.doi.org/10.1016/0191-2615(79)90022-5.

[85] Éva Tardos and Tom Wexler. Network formation games and the potential function method. In Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani, editors, *Algorithmic Game Theory*, chapter 19. Cambridge University Press, 2007. ISBN 978-0-521-87282-9. Available also via http://theory.stanford.edu/~tim/.

[86] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006. URL http://dx.doi.org/10.1007/s10107-004-0559-y.

[87] John G. Wardrop. Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers, Part II*, 1(3):325–362, 1952.

[88] Alison Watts. A dynamic model of network formation. *Games and Economic Behavior*, 34(2):331–341, 2001. URL http://dx.doi.org/10.1006/game.2000.0803.

[89] Andreas Wächter. Short tutorial: Getting started with Ipopt in 90 minutes. Technical Report RC24790, IBM Research Division, 2009. URL http://domino.watson.ibm.com/library/CyberDig.nsf/papers/22B21525709E4BFB852575A700587AF0.

[90] Hai Yang, Deren Han, and Hong Kam Lo. Efficiency of atomic splittable selfish routing with polynomial cost functions. *Networks and Spatial Economics*, 8(4):443–451, 2008. URL http://dx.doi.org/10.1007/s11067-007-9017-8.

Colophon

This thesis was written using the LaTeX typesetting system and the `book` document class. 11pt Palatino font was used with `\linespread{1.05}` on Crown Quarto paper. The text block was 11.8 cm wide and 18.2 cm high (without running titles). The margin ratio was 4 : 3, this means having somewhat larger inner than outer margins. Wide figures, equations, and tables were allowed to hang into the inner margin (in the appendix also into the outer margin). There is also an electronic version available that implements a more symmetrical left-right layout, which might be better suited for screen viewing or printing on 'normal' paper, such as A4.

Figures were created with Ti*k*Z and plots were done using the R System; a few plots were also done using Ti*k*Z and Gnuplot. An almost endless number of additional helpful LaTeX packages was used, provided by the TeX Live project. In particular, to name a few, it was relied on the AMS packages, and the `algorithm2e`, `babel`, `booktabs`, `boolexpr`, `calc`, `caption`, `changepage`, `csquotes`, `courier`, `dsfont`, `enumitem`, `eso-pic`, `etex`, `fancyhdr`, `floatrow`, `fontenc`, `geometry`, `graphicx`, `grffile`, `hyperref`, `inputenc`, `mathpazo`, `mathtools`, `microtype`, `multirow`, `natbib`, `nicefrac`, `numprint`, `sectsty`, `svninfo`, `units`, `url`, `varioref`, and `xifthen` package.

Source code was written using the Vi IMproved editor. Source code was managed using the Subversion and Mercurial revision control systems.